Department of Electrical and Computer Engineering
Faculty of Technical Sciences Aarhus University

Blackleg Risk Detection in Industrial Potato Fields Using UAV-Based RGB Imagery and Deep Learning

Master's Thesis

Blackleg Risk Detection in Industrial Potato Fields Using UAV-Based RGB Imagery and Deep Learning

Master's Thesis
January, 2026
By
Emil Hilligsøe Lauritsen

Cover photo: AU Photo and Denis Selnihhin
Published by: Aarhus University, Denmark
www.au.dk

Preface

This thesis was written as part of the Master's programme in Computer Engineering at Aarhus University, in collaboration with AU Agro. I would like to express my sincere gratitude to everyone who supported and guided me throughout the completion of this thesis.

First and foremost, I am deeply grateful to my supervisor, Cláudio Gomes, for his guidance, encouragement, and constructive feedback throughout the project. I am also very thankful to my co-supervisor, Davide Cammarano, for his valuable insights, advice, and engagement during the research process.

I would further like to thank the expert farmer Niels Lauritsen for his assistance with annotating the imagery and for sharing practical insights that helped ensure the practical relevance of the final work. Lastly, I thank Marco Canicatti for his support in capturing the UAV data used in this study.

Abstract

Blackleg is a seed-borne bacterial disease that occurs worldwide and can lead to significant yield losses in seed potato production. Currently, blackleg is primarily detected through manual field inspection, a process that is time-consuming and subjective. This thesis proposes a machine learning framework for automated detection of blackleg risk areas using Unmanned Aerial Vehicle (UAV)-based Red–Green–Blue (RGB) imagery.

Data are collected using a UAV and processed into an orthomosaic, which is annotated into *Risk* and *No-Risk* areas by an expert farmer. The orthomosaic is divided into image tiles that serve as input to machine learning models. The resulting dataset exhibits a strong class imbalance. Experimental results show that, for this dataset, the imbalance is most effectively handled using a weighted sampler in combination with Cross-Entropy (CE) loss.

Three model architectures are evaluated: a custom Convolutional Neural Network (CNN) designed specifically for risk detection, two ResNet-18 models trained either from scratch or fine-tuned using ImageNet-1K pretrained weights, and a vision transformer (ViT-S/16) pretrained on ImageNet. The results demonstrate that blackleg risk detection from UAV-based RGB imagery is feasible under industrial field conditions. On held-out spatial test data, the best-performing model achieves an $F1_{risk}$ score of 0.46, confirming that meaningful risk signals can be detected despite subtle visual symptoms and severe class imbalance. No single architecture dominates across all evaluation metrics; instead, clear trade-offs are observed. The custom CNN achieves the most balanced overall performance, while ImageNet-pretrained models exhibit higher recall for the minority risk class on this dataset. These findings establish a practical baseline for future work on scalable, UAV-based disease risk detection in seed potato production.

Resumé

Sortben er en frøbåren bakteriel sygdom, der forekommer globalt og kan medføre betydelige udbyttetab i produktionen af læggekartofler. I dag opdages sortben primært gennem manuel markinspektion, hvilket er en tidskrævende og subjektiv proces. Dette speciale foreslår et maskinlæringsbaseret framework til automatisk detektion af sortben-risikoområder ved hjælp af UAV-baserede RGB-billeder. Data indsamles ved hjælp af en UAV og behandles til et ortomosaik, som annoteres i *Risiko-* og *Ikke-Risiko*-områder af en ekspertlandmand. Ortomosaikken opdeles efterfølgende i billed-tiles, som anvendes som input til maskinlæringsmodeller. Det resulterende datasæt er præget af en markant klasseubalance. De eksperimentelle resultater viser, at denne ubalance for dette datasæt håndteres mest effektivt ved brug af en vægtet sampler i kombination med cross-entropy-loss. Tre modelarkitekturer evalueres: et specialdesignet Convolutional Neural Network (CNN) udviklet specifikt til risikodetektion, to ResNet-18-modeller trænet enten fra bunden eller finjusteret ved hjælp af ImageNet-1K-prætrænede vægte samt en Vision Transformer (ViT-S/16) prætrænet på ImageNet. Resultaterne viser, at detektion af sortben-risiko ud fra UAV-baserede RGB-billeder er mulig under industrielle markforhold. På et holdt-ude spatialt testdatasæt opnår den bedst præsterende model en $F1_{risk}$-score på 0.46, hvilket indikerer, at meningsfulde risikosignaler kan identificeres trods subtile visuelle symptomer og alvorlig klasseubalance. Ingen enkelt arkitektur dominerer på tværs af alle evalueringsmetrikker; i stedet observeres tydelige trade-offs. Det specialdesignede CNN opnår den mest balancerede samlede performance, mens ImageNet-prætræning er associeret med højere recall for den sjældne risikoklasse. Disse resultater etablerer et praktisk udgangspunkt for fremtidigt arbejde med skalerbar, UAV-baseret sygdomsrisikodetektion i produktionen af læggekartofler.

Title Page

Title:
Blackleg Risk Detection in Industrial Potato Fields Using UAV-Based RGB Imagery and Deep Learning

Author:
Emil Hilligsøe Lauritsen, Student Number: 202004154

Project Period:
1st of September 2025 – 3th of January 2026

Supervisor:
Cláudio Ângelo Gonçalves Gomes
Co-supervisors:
Davide Cammarano

Department of Electrical and Computer Engineering
Faculty of Technical Sciences, Aarhus University
http://www.ece.au.dk

Contents

Chapter 1

Introduction

Between 2004 and 2024, the cultivated area of seed potatoes in Denmark increased from 5,079 to 9,645 hectares, corresponding to a growth of approximately 90% [1]. To ensure seed potato quality, official field inspections are conducted to identify diseased plants. Due to the strict quality requirements, the detection of even a single diseased plant can result in declassification of an entire production lot. In particular, the presence of blackleg, caused by *Pectobacterium atrosepticum*, is sufficient to trigger declassification, leading to substantial financial losses for farmers [2].

Blackleg is a seed-borne bacterial disease that occurs worldwide. The pathogen typically originates from infected seed potatoes and can spread across multiple generations, which underlines the importance of removing infected plants at an early stage. Symptoms may initially be absent or appear as small, water-soaked lesions at the base of the stem. In advanced stages, the disease can cause extensive basal rot, wilting, and eventually plant death [3]. Figure 1.1 illustrates typical blackleg symptoms.

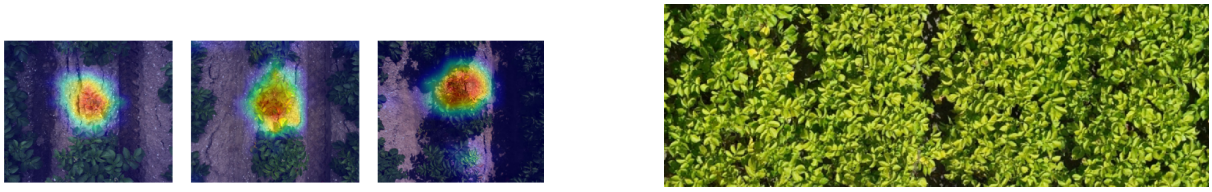
Figure 1.1: Examples of blackleg symptoms in potato plants. The red circle highlights a characteristic blackened stem base in one example. Overall, the figure illustrates the large visual variability of blackleg symptoms, ranging from stem rot to more subtle plant stress, which complicates automated detection. [4]

Currently, blackleg is primarily detected through manual field inspection, a time-consuming and subjective process that also risks disease spread, highlighting the usefulness of automated and scalable detection systems.

However, developing such systems is challenging. A common problem in applied precision agriculture and anomaly detection is class imbalance, as non-risk areas are vastly

more frequent than risk areas. This imbalance makes the identification of diseased plants statistically and computationally more difficult, and must therefore be accounted for when designing detection algorithms. [5]

Previous studies have demonstrated that potato diseases can be detected using RGB or multi-spectral imagery from UAVs, handheld cameras, or vehicle-mounted systems. However, most prior work has relied on data collected from research plots, which are less complex than industrial fields where higher plant density poses additional challenges. [4] [6]. This is highlighted in figure 1.2. Furthermore, no studies have explored UAV-based imagery for blackleg risk detection.



(a) Research plot

(b) Industrial field

Figure 1.2: Visual comparison of a research plot [4] (left) and an industrial potato field (right). The research plot exhibits more controlled planting patterns and lower plant density, whereas the industrial field is denser and more heterogeneous, making automated blackleg risk detection more challenging.

This thesis addresses the lack of studies on UAV-based blackleg risk detection under realistic industrial field conditions. To our knowledge, it presents one of the first evaluations of blackleg risk detection using UAV-based RGB imagery collected from an industrial seed potato field.

The study investigates the feasibility of detecting blackleg risk areas despite subtle visual symptoms and severe class imbalance, and examines the impact of different class imbalance handling strategies and model architectures. The results indicate that sampling-based imbalance handling using CE loss combined with a weighted sampler provides robust performance for the considered dataset. Furthermore, the comparison of architectures highlights trade-offs between task-specific and pretrained models, where a custom CNN yields more balanced performance, while ImageNet pretraining primarily benefits recall for the minority risk class.

Together, these findings establish a practical baseline for future work on scalable blackleg risk detection systems that can be used to prioritize manual field inspections in seed potato production.

1.1 Research questions

To structure the investigation and guide the experimental design, we pose the following RQ's.

- **Research Question (RQ)1:** Can a model be trained to detect *blackleg risk areas* from UAV imagery?
- **RQ2:** How should class imbalance be handled to maximise $F1_{risk}$ for the minority (risk) class?
- **RQ3:** Which model architecture performs best for this task? Specifically, how do a simple CNN, a pretrained Residual Neural Network (ResNet), and a Vision Transformer (ViT) compare?
- **RQ4:** What is the effect of ImageNet pretraining on accuracy and $F1_{risk}$ in blackleg risk detection?

1.2 Case Study

To investigate the research questions, we conducted a case study using data collected from an industrial potato field in northwestern Denmark in July 2025, where late-stage blackleg was known to be present.

A DJI Zenmuse P1 camera, mounted on a DJI Matrice 300 UAV, was used to capture imagery in the RGB spectrum. One flight was conducted at an altitude of 12 m. The dataset was manually annotated with assistance from an expert farmer.

We chose to detect blackleg *risk areas* rather than the blackleg infection directly, as the spatial resolution was insufficient for annotators to reliably identify infected plants.

Following data acquisition, the imagery was processed into an orthomosaic. From the mosaic, image tiles were extracted to construct a dataset suitable for machine-learning experiments.

This case study forms the foundation for the subsequent model development and evaluation presented in this thesis.

Chapter 2

State of The Art

Traditionally, blackleg in potatoes is detected by a domain expert, typically a farmer or an agricultural consultant. The expert enters the field on foot or using a small vehicle and slowly traverses the crop rows. When a plant exhibits potential symptoms, the expert stops and performs a closer inspection. If the plant is deemed infected, it is removed immediately to limit further spread.

This process is highly time-consuming, as each field must be inspected multiple times throughout the growing season. Repeated inspections are necessary because the disease becomes easier to detect at later growth stages. At the same time, the tolerance threshold for avoiding declassification is extremely strict, the number of blackleg-infected plants permitted in certified seed potato fields is 0% [2].

2.1 Machine Learning and Plant Disease Detection

Machine learning has become a key tool in plant disease detection by enabling automated analysis of large-scale imagery and the identification of subtle visual patterns that are difficult to assess consistently through manual inspection. The growing use of UAVs has further advanced this field by enabling high-resolution and field-scale imaging. Depending on the sensor payload, UAV platforms can capture both visual and spectral information relevant to early plant stress.

Earlier approaches relied on handcrafted features combined with classical classifiers, requiring manual feature engineering tailored to specific imaging conditions. In contrast, modern deep-learning methods learn features directly from raw imagery and have therefore become the dominant paradigm in agricultural disease detection. While a broad range of crops and diseases have been studied using these methods, relatively few works focus on potato diseases, and only a limited number address blackleg in particular. This is especially true for studies conducted under industrial field conditions, motivating the focus of this work.

2.2 Blackleg detection

To our knowledge, only Hulsman (2025) and Alfonso et al. (2019) have examined blackleg classification, and both rely on ground-based imaging collected in research fields rather than industrial potato fields. No existing work has explored the use of UAV imagery for blackleg detection, nor has any study addressed the problem of identifying blackleg risk areas. This gap highlights the need for methods capable of operating under realistic agricultural conditions.

G.W. Hulsman (2025) [4] uses EfficientNetV2 to detect blackleg in potatoes based on a dataset collected over several years and across multiple potato varieties. The resulting dataset contains 4,528 images and is balanced between healthy and symptomatic plants. Images were acquired using a modified cart-mounted camera system. The model achieves strong performance, with a recall of 81% and a precision of 91%. A key strength of the study is its robustness across years and varieties, indicating good generalization to temporal and biological variation.

Alfonso et al. (2019) [6] similarly investigate blackleg detection using RGB imagery. Their dataset consists of 532 images collected over several days in a research field using a tractor-mounted imaging system with controlled lighting. Images were selected to ensure plant isolation and to exclude trivial cases. The authors evaluate ImageNet-pretrained ResNet-18 and ResNet-50 models, where ResNet-18 achieves an precision of 0.95 and a recall of 0.91 for the blackleg class, outperforming the deeper ResNet-50.

In summary, these studies demonstrate that deep CNNs can achieve high accuracy for blackleg detection under controlled conditions. However, both rely on data collected in research fields with isolated plants and controlled lighting, which differ substantially from industrial potato fields. Moreover, the datasets used exhibit a much higher prevalence of infected plants than is typically observed in practice. In industrial seed potato fields, blackleg cases are often rare, requiring models to operate under severe class imbalance and to prioritize high recall for the minority (risk) class. Consequently, existing studies provide limited insight into how well these methods generalize to realistic field-scale conditions.

2.3 General work on Potato diseases

Other studies have investigated the use of UAV and multi-spectral imagery to detect potato diseases such as late blight and Potato Virus Y, and while targeting different diseases, they provide transferable insights into how disease detection can be approached.

Franceschini et al. (2019) [7] used UAV-mounted hyperspectral sensors to detect early spectral changes in potato plants infected with late blight, achieving detection even at very mild infection levels (2.5–5% leaf area). The study highlights the strong potential of hyperspectral UAV imaging for early, field-scale disease monitoring. A detailed description of their workflow is provided in Appendix A.1.

Polder et al. (2019) detect Potato Virus Y in an research seed potato field using a hyperspectral camera mounted in a vehicle-based imaging box. Their model achieves strong performance, with precision above 0.78 and recall 0.88 for the infected plant, demonstrating the potential of hyperspectral imaging for reliable, field-scale disease detection. [8]

Duarte-Carvajalino et al. (2018) [9] investigated the use of a low-cost multi-spectral camera mounted on a UAV to estimate late blight severity percentage in potatoes.The sensor employed a blue–green–near-infrared filter, and the authors evaluated several machine-learning models. These include multilayer perceptrons, CNNs, Support Vector Machine (SVM) regression, and random forest regression, treating disease severity as a regression task. Their results showed that the best-performing model was a CNN with four convolutional layers, achieving a mean absolute error of 11.72%. This architecture is computationally efficient and inexpensive to deploy, making it attractive for practical field use.

Biswas et al. (2016) [10] used a classical pipeline combining Fuzzy C-Means segmentation with a multilayer perceptron to assess potato blight severity in high resolution RGB images. Although the method achieved 0.93 accuracy, it relied on only 27 farmer-captured images. A detailed description is provided in Appendix A.1.

The studies above highlight the potential of using UAVs for potato disease detection, with most relying on multispectral or hyperspectral imaging to enhance detection performance. While these works demonstrate the promise of UAV-based approaches, they are primarily conducted in controlled or research field settings and depend on specialized sensors, leaving open the question of how well disease detection can be achieved in industrial potato fields using standard RGB imagery.

2.4 Vision Models

To our knowledge, very few vision models have been applied to potato disease detection, despite the fact that they have been shown to equal or outperform CNNs in a wide range of other object detection tasks. Espitalier et al. [11] employ a Bidirectional Encoder representation from Image Transformers (BEiT)-based Vision Transformer for plant-species detection, leveraging extensive multi-stage pretraining on ImageNet and the large Pl@ntNet dataset. This strong domain-specific pretraining enables effective detection even under severe class imbalance, though at high computational cost. Details are provided in Appendix A.1.1. Parez et al. (2023) [12] propose GreenViT, a Vision Transformer architecture for plant disease detection that achieves state-of-the-art performance across multiple benchmark datasets. The model reports classification accuracies of 0.97 on the PlantVillage dataset, 0.94 on the Leaf Images Data Repository, and 0.96 on the Plant Composite dataset, demonstrating that Vision Transformers can effectively capture subtle visual symptoms in plant imagery. A detailed description is provided in Appendix A.1.1.

Chapter 3

Methods and Tools

This chapter describes the tools and methods used to investigate the research questions defined in Section 1.1. The chapter is structured as follows.

First, the data collection procedure and preprocessing steps used to prepare the data for machine learning are described. The resulting dataset, including its structure and key characteristics, is then presented. Next, the machine learning architectures evaluated in this work are introduced. Finally, the class imbalance handling strategies investigated are described.

3.1 Data collection and processing

Figure 3.1 provides an overview of the data processing pipeline, illustrating the steps required to transform raw UAV imagery collected in the field into a dataset suitable for machine learning.

The UAV data acquisition and generation of orthomosaics were carried out in collaboration with Marco Canicatti and AU Agro. Following data collection, expert-driven polygon annotations were created to identify blackleg risk areas within the field. These annotations form the basis for assigning risk labels to individual image tiles used during model training and evaluation.

The geometric preprocessing steps, including image rotation and cropping, are described in detail in Appendix B.

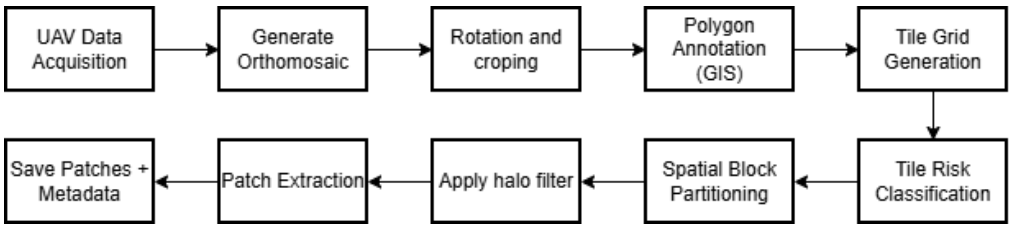


Figure 3.1: Overview of the data collection and processing pipeline.

3.1.1 Annotation by expert farmer

To obtain accurate annotations of blackleg risk areas in the field, we collaborated with a farmer who has more than thirty years of experience in identifying diseases in potato crops. His domain expertise ensured that the labeled regions reflected real agronomic knowledge rather than purely visual interpretation, thereby strengthening the reliability of the ground-truth dataset.

To create annotation polygons, we used QGIS.[1] Using QGIS enabled us to perform the annotations prior to tile generation, which made the process considerably faster compared to manually annotating individual tiles. Figure 3.2 illustrates the annotation polygons, shown in orange, overlaid on the orthophoto.

Figure 3.2: Illustration of the data preparation pipeline. An RGB orthomosaic with expert-annotated blackleg risk polygons (orange) is first divided into overlapping tiles (white) using a fixed stride, resulting in partial tile overlap. Tiles are then grouped into larger spatial blocks (dark blue) to prevent spatial leakage between training, validation, and test splits. Tiles located within a margin (red) near block boundaries are discarded to avoid overlap across splits.

To improve the annotator's ability to identify risk areas, we enabled the red measurement grid shown in the top-right corner of Figure 3.2 and rotated the orthomosaic so that the field rows were horizontally aligned. This orientation provides a viewpoint similar to what a farmer experiences when walking/drives though the field.

The annotator was instructed as follows: *"Label the blackleg risk areas that you would inspect more closely if you were standing in the field yourself."*

[1] `https://qgis.org/project/overview/`

3.1.2 Tile size

As part of the tile grid creation, both a tile size and a stride must be chosen. The tile size determines the maximum spatial context available to the model in each sample: larger tiles include broader spatial structure, while smaller tiles restrict the model to more localized visual information.

The stride specifies the step size between consecutive tiles. When the stride is smaller than the tile size, the resulting spatial overlap increases the number of training samples while ensuring that neighboring tiles are still distinct. This overlap also improves the likelihood that important features are fully captured within at least one tile, rather than split across tile boundaries.

To guide this choice, we computed the median size of the annotated polygons, which was approximately 129×129 pixels. Using a tile size of 224×224 pixels therefore ensures that, in most cases, an entire annotated region can be captured within a single tile. We set the stride to 128 pixels to increase the likelihood that at least one tile fully contains the complete feature of interest.

A further practical motivation for choosing a 224×224 input size is compatibility with widely used pretrained architectures such as ResNet and ViT, which are typically trained on this image size. Using a standard tile size simplifies fine-tuning, ensures that pretrained weights behave as intended.

3.1.3 Classification of tiles

To classify the tiles in the grid, all annotation polygons were first merged into a single unified polygon layer. For each tile, the area of intersection with this merged polygon was computed. A tile was labeled as a *Risk* area if the intersection exceeded 5% of the tile's total area. The 5% overlap threshold represents a practical compromise. A lower threshold reduces the risk of false negatives by ensuring that tiles capturing only a partial extent of a diseased region are still included as positive samples. However, it also increases the difficulty of the classification task, as some positive tiles may contain only weak disease signals relative to background vegetation. Furthermore, lowering the threshold increases the number of positive samples, which may introduce label noise and reduce class separability. Despite these trade-offs, the selected threshold provides a reasonable balance between reducing class imbalance and maintaining meaningful visual evidence of disease within positive tiles.

3.1.4 Spatial block partitioning

After classifying the tiles in the grid, the dataset was divided into spatial blocks as, seen on figure 3.2 (dark blue). This is to prevent data leakage caused by overlapping tiles when splitting into training, validation, and test sets. After defining the blocks, a margin

(red) equal to the overlap was removed around each block to ensure that no two blocks contained overlapping or shared image regions. This is visualized in the bottom of figure 3.2

This approach introduces a trade-off: on one hand, we need a sufficient number of blocks to enable flexible splits between training, validation and testing; on the other hand, increasing the number of blocks also increases the amount of data lost due to the dropped margins. To balance this, a block size equal to eight times the patch size was chosen, a compromise between clean separation and minimal data loss.

3.2 Dataset

To create a realistic evaluation setup, we partitioned regions of the field into a training region and a separate test region. The test set corresponds to a continuous block of tiles taken from the top section of the orthomosaic in 3.2 (blocks 0–9). Evaluating the model on a spatially continuous area ensures that the test data truly represent unseen parts of the field, rather than isolated scattered samples.

A consequence of this design is that the class distribution in the test set is determined entirely by the natural conditions in that region. In other words, we do not artificially balance or modify the test split; we use the data as they appear in the field.

In total, the dataset contains 12,078 samples across all spatial blocks. As expected the dataset is highly imbalanced, with 11,462 tiles labeled as *No Risk* (94.9%) and 616 labeled as *Risk* (5.1%).

When divided into training and test splits, this imbalance becomes even more apparent. The training set contains 11,472 *No Risk* tiles (95.9%) and 486 *Risk* tiles (4.1%), while the test set contains 1,005 *No Risk* tiles (87.2%) and 147 *Risk* tiles (12.8%). The higher proportion of risk samples in the test region introduces a slightly more challenging evaluation scenario, but it also provides a more realistic assessment of the model's ability to handle varying disease levels across different parts of the field.

If we inspect Figure 3.3, we observe several examples of tiles classified as risk areas. In each of them, there is clear evidence of a plant lying down, a typical symptom of blackleg, where the stem collapses due to infection. However, similar visual patterns can also occur for unrelated reasons, such as mechanical damage or natural growth variations. This highlights the difficulty of the problem: visual cues associated with blackleg are not always unique to the disease. Index 11631 also highlights another issue: an artifact introduced during the orthomosaic stitching process has distorted the tile. Such distortions do not reflect the real appearance of the field and therefore produce samples that are unsuitable for machine learning, as they may mislead the model during training or evaluation.

Figure 3.3: Example of tiles classified as *Risk*

If we inspect Figure 3.4, we observe several examples of *No Risk* tiles. These tiles all contain healthy potato plants with no visible symptoms of blackleg.

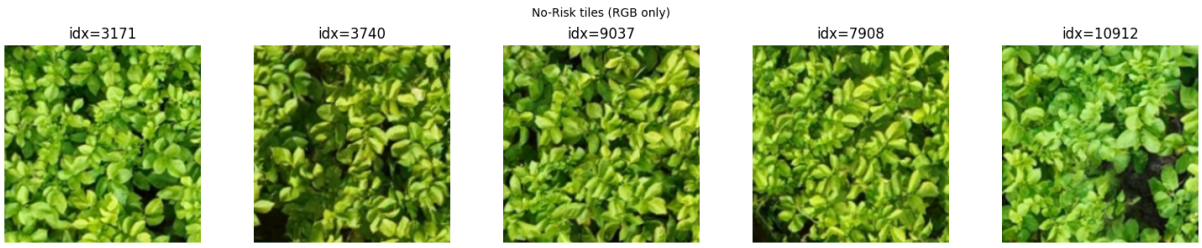


Figure 3.4: Example of tiles classified as *No Risk*

3.3 Machine Learning Architectures

The following section describes the machine learning architectures evaluated in this work. We consider three model families: a convolutional neural network, a ResNet model, and a Vision Transformer. These architectures provide a diverse set of inductive biases and represent both classical convolutional approaches and modern transformer-based methods.

3.3.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are widely used for image analysis due to their ability to learn spatially localized features through convolutional filters with shared weights. A key property of CNNs is shift invariance, which allows the same visual pattern to be detected regardless of its spatial location in the input image. This is particularly relevant for risk-area detection, where disease symptoms may occur anywhere within a tile [13].

CNNs further exploit the strong correlation between neighboring pixels by restricting the receptive field of early layers, enabling the learning of fine-grained local patterns such as texture and shape. While deeper layers can learn increasingly abstract representations, early-stage feature extraction remains important when target cues are subtle and spatially localized. Since blackleg-related visual symptoms are expected to manifest as local deviations in plant appearance rather than globally distributed patterns, CNN-based architectures provide a suitable inductive bias for the problem setting considered in this work.

A more detailed discussion of the historical development of CNNs and their mathematical formulation is provided in Appendix A.2.

3.3.2 Residual Neural Networks

ResNets were introduced to enable effective training of deep convolutional architectures by addressing optimization difficulties that arise as network depth increases [14]. Their defining feature is the use of identity shortcut connections, which allow information and gradients to bypass convolutional layers, improving optimization stability and convergence. ResNets have demonstrated strong performance across a wide range of image classification tasks and are commonly used in transfer learning settings. A characteristic design choice in standard ResNet architectures is early downsampling, which quickly reduces spatial resolution while encouraging the network to learn more abstract feature representations. This is generally beneficial in large-scale recognition tasks, where robustness and semantic abstraction are important. For blackleg risk detection from UAV imagery, this introduces a potential trade-off. Early downsampling may weaken very fine-grained spatial details, but it may also encourage the model to rely on higher-level representations formed by aggregating multiple low- and mid-level visual cues across space. As a result, it is not clear in advance whether this design choice will be beneficial or detrimental for this specific task. For this reason, ResNets are included in this study to systematically evaluate how deeper, pretrained architectures compare to a simpler CNN when detecting blackleg risk areas. A more detailed discussion of the historical development of ResNet architectures and the underlying mathematical formulation of residual learning is provided in Appendix C.2.

3.3.3 Transfer Learning

Research has shown that the first layers of deep CNN models typically learn simple visual patterns such as edges and basic color regions. [15]. These low-level representations are largely universal across visual domains, meaning that the same features are useful for many different image recognition tasks. Yosinski (2014) [15] demonstrated that while higher layers become task-specific, the early convolutional layers capture general visual primitives that can be effectively transferred between datasets. CNNs are inherently data-hungry, and limited training data can lead to overfitting when models are trained from scratch, resulting in poor generalization to unseen data. In the context of this study, the amount of available training data is small and cannot be expanded further at this stage. Transfer learning therefore enables the use of high-capacity architectures without overfitting. Pretraining on datasets such as ImageNet supplies general visual representations, while fine-tuning allows later layers to adapt to domain-specific patterns present in UAV imagery.

To apply transfer learning, fine-tuning strategies are commonly employed in which selected layers of a network are updated during training. A typical approach is to freeze

early convolutional layers and fine-tune only the later, more task-specific layers together with the classification head. For the ResNet architecture, this corresponds to keeping the initial convolutional blocks fixed while fine-tuning the final residual blocks and the classifier.

The appropriate fine-tuning depth depends on the similarity between the source and target domains. Given the substantial differences between ImageNet images and UAV imagery of potato fields, allowing the later convolutional blocks to adapt is expected to improve performance while retaining the benefits of pretrained representations.

3.3.4 Vision transformers

Vision Transformers (ViTs) were introduced by Dosovitskiy et al. in "An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale" [16]. The key idea is to adapt the transformer architecture to visual data by representing an image as a sequence of fixed-size patches. An input image is divided into non-overlapping patches, each of which is flattened, linearly embedded, and augmented with positional information. These patch embeddings are processed by a transformer encoder using self-attention, allowing the model to capture relationships between all patches in the image. A dedicated classification token aggregates the information and is used to produce the final prediction. This is described in depth in Appendix C.3

A key advantage of ViTs is their ability to model global context through self-attention, allowing each image patch to directly interact with every other patch. This enables the capture of long-range dependencies that are difficult to model efficiently with standard CNNs, which rely on progressively expanding the network depth which then expands the receptive fields. In the context of blackleg risk detection, this property is potentially beneficial. Field inspectors often identify risk by spotting plants that stand out relative to their surrounding canopy. Similarly, while local discolorations and texture changes are important, risk may also manifest as relative anomalies. ViTs can explicitly compare patches across the entire tile. However, transformer-based models are typically more data-hungry than CNNs and often require large-scale pretraining to perform well [17]. For this reason, transfer learning is essential when applying ViTs to datasets of limited size.

3.4 Class imbalance methods

In section 3.2 we describe the large class imbalance present in the data. The following section will describe methods which we are going to use to address the issue.

To summarize, we investigate the effect of the following tools for handling class imbalance and training stability:

- **Sampler type:** weighted or random

- **Loss function:** CE or focal loss
- **Bias initialization:** enabled or disabled
- **Label smoothing:** applied with $\epsilon \in [0, 0.2]$

Several of these strategies interact with or counteract each other. For example, weighted sampling changes the effective class distribution seen by the model, which may conflict with class weights, bias initialization, or focal loss. To avoid such interference and to make the evaluation interpretable, we group the experiments into four coherent strategies. Section 3.4.1 describes the four strategies and section 3.4.2-3.4.8 describes the imbalance tools applied in the strategies.

3.4.1 Class imbalance strategies

The four proposed strategies are:

Random Sampling + CE loss - Baseline This strategy uses a random sampler and standard CE loss, with bias initialization disabled. This provides a tool-free baseline and allows us to isolate the effect of imbalance-handling methods.

Random Sampling CE loss + Smoothing In this strategy we use a random sampler and standard CE loss, and enable label smoothing to stabilize training. Bias initialization remain disabled, ensuring that the class distribution is not altered and that smoothing can be evaluated independently of other imbalance-handling methods.

Weighted Random Sampling + CE Loss This strategy enables the weighted sampler while keeping CE loss. We disable label smoothing, class weights, and bias initialization, as these may conflict with the altered batch distribution introduced by weighted sampling. This configuration isolates the effect of sampling-based imbalance correction.

Random Sampling + Focal loss In this strategy we investigate focal loss using a random sampler, ensuring the natural batch distribution is preserved. Bias initialization is enabled as recommended by Lin et al. [18]. This configuration allows us to study the behavior of focal loss under the original class imbalance.

3.4.2 Sampler types

We first investigate the `WeightedRandomSampler` available in PyTorch as a data-level strategy for handling class imbalance during training. This sampling approach has been used in prior deep learning studies [19]. The sampler assigns a sampling probability to each tile inversely proportional to its class frequency, resulting in mini-batches that contain

approximately equal proportions of *Risk* and *No-Risk* samples. In practice, this means that minority *Risk* tiles are sampled more frequently, sometimes multiple times per epoch, while all majority-class tiles are still encountered across training epochs.

Although this strategy increases the model's exposure to minority samples, it also introduces a potential risk of overfitting, as the same minority tiles may be repeatedly presented without substantial variation. The method is computationally efficient and preserves the full dataset, but it alters the effective class distribution observed during training. As a consequence, loss-based imbalance handling strategies such as class weighting, focal loss, or prior bias initialization must be applied with caution, since their behavior depends on the assumed training distribution and may interact with the sampling strategy.

The `RandomSampler`, which is the standard choice in PyTorch when no sampler is specified, samples uniformly from the dataset without adjusting for class imbalance. This means that the mini-batches will naturally reflect the skewed class distribution. While this avoids the potential overfitting introduced by repeated minority samples, it also means the model receives far fewer updates driven by the rare "Risk" class. In highly imbalanced settings like ours, this often results in a model that learns to prioritize overall accuracy over minority-class recall, making the WeightedRandomSampler a useful alternative despite its trade-offs.

3.4.3 Loss functions

CE serves as our primary baseline loss. For binary classification, we define the probability assigned to the true class as

$$p_i = \begin{cases} \hat{p}_i, & \text{if } y_i = 1, \\ 1 - \hat{p}_i, & \text{if } y_i = 0, \end{cases}$$

where $\hat{p}_i = \sigma(z_i)$ is the sigmoid-transformed logit. Using this notation, the CE loss becomes:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{N} \sum_{i=1}^{N} \log(p_i).$$

Intuitively, if the model assigns a high probability to the correct class, p_i is close to 1 and the loss $-\log(p_i)$ becomes very small. If the model is unsure or assigns a low probability to the true class, p_i becomes small and the loss grows rapidly. A key limitation of standard CE loss in highly imbalanced settings is that the overall loss becomes dominated by a large number of well-classified majority-class samples, while misclassified minority-class samples contribute relatively little to the gradient.

To mitigate this limitation in settings with severe class imbalance, we evaluate `focal loss`, proposed by Lin et al. for dense object detection in RetinaNet [18]. Focal loss down-weights well-classified examples and focuses the learning process on hard, misclassified

samples, making it particularly suitable for datasets dominated by a large number of negative examples.

Using the same definition of p_i as in CE, focal loss is defined as:

$$\mathcal{L}_{\text{focal}} = -\alpha_t (1 - p_i)^{\gamma} \log(p_i),$$

where α_t balances the contribution of each class and γ controls how strongly easy examples are down-weighted.

When $\gamma = 0$, focal loss reduces to standard CE with class weights α_t. As γ increases, the factor $(1 - p_i)^{\gamma}$ suppresses the loss contribution from well-classified examples (where p_i is high) and places greater emphasis on hard or misclassified samples.

The term α_t allows explicit control over class imbalance: assigning a larger value to the minority class increases its influence during training, while a smaller value reduces the contribution of the majority class. Thus, γ focuses learning on difficult samples, while α_t ensures that each class contributes proportionally to the overall loss.

In our experiments, γ is treated as a hyperparameter and optimized to evaluate whether focal loss provides measurable benefits over CE in detecting the minority "Risk" class.

3.4.4 Bias Initialization

Following Lin et al. [18], we also initialize the classifier bias using the empirical class priors. Given class frequencies n_c, we compute the prior $\pi_c = n_c / \sum_j n_j$ and set the bias of the final linear layer to $\log(\pi_c)$. This ensures that, at the start of training, the model predicts class probabilities consistent with the dataset distribution rather than a uniform 50/50 split. Such prior bias initialization stabilizes early training under heavy class imbalance by preventing the majority class from dominating the initial gradients.

3.4.5 Label Smoothing

When classifying tiles in Section 3.1.3 using a hard threshold of 5%, images with a 5.1% intersection are labeled as "Risk", while those with a 4.9% intersection are labeled as "No Risk", despite being visually and contextually very similar. This introduces uncertainty near the decision boundary, which can confuse the model.

To mitigate this issue, we apply label smoothing. In binary classification, ground-truth labels are typically represented as hard targets $y \in \{0, 1\}$. With label smoothing, these hard labels are replaced by softened targets that distribute a small portion of the probability mass to the opposite class. For a smoothing factor ε, the smoothed targets become:

$$\tilde{y} = \begin{cases} 1 - \varepsilon, & \text{if the true label is "Risk" (1)} \\ \varepsilon, & \text{if the true label is "No Risk" (0)} \end{cases}$$

Thus, instead of training the model to assign full confidence to one class (e.g. $\tilde{y} = 1$ for risk), the target becomes slightly softened (e.g. $\tilde{y} = 0.9$ for $\varepsilon = 0.1$). This reduces the tendency of the model to become overconfident and encourages better generalization, particularly for tiles close to the 5% intersection threshold where the ground-truth label is inherently uncertain. Although label smoothing does not directly address class imbalance, it mitigates its effects by preventing overconfident predictions on the majority class, thereby allowing minority-class samples to exert greater influence on the decision boundary.

3.4.6 Data Augmentation and Normalization

To improve generalization and reduce overfitting, basic data augmentations are applied to the training images. These include random horizontal and vertical flips, random rotations, and mild color jittering in brightness, contrast, saturation, and hue.

Since the ResNet and Vision Transformer (ViT) models used in this work are pretrained on ImageNet, input images are normalized using ImageNet mean and standard deviation statistics. This normalization aligns the input distribution with that seen during pretraining, stabilizes optimization, and facilitates more effective reuse of pretrained features, thereby improving training convergence and overall performance.

3.5 PyTorch and Optuna

All deep learning models in this work are implemented using the PyTorch framework. PyTorch provides a flexible and efficient platform for constructing neural network architectures, handling automatic differentiation, and training models on GPU hardware. Its modular design enables seamless integration of custom loss functions, sampling strategies, and experimental configurations used throughout this thesis [20].

Hyperparameter optimization is performed using the Optuna framework. Optuna provides a flexible interface for defining complex search spaces, integrating custom objective functions, and managing large-scale hyperparameter optimization. It integrates naturally with PyTorch, allowing model architectures, training loops, and evaluation metrics to be embedded directly within each optimization trial [21].

Chapter 4

Model Design and Implementation

In this chapter, we expand on the models introduced in Section 3.3. We describe the implementation choices made for each architecture, including considerations related to model size, depth, and overall design. Table 4.1 lists the models evaluated in this work along with their parameter counts. These values provide a rough indication of model capacity and computational cost.

Table 4.1: Model complexity in terms of total and trainable parameters for the architectures evaluated in this work.

Model	**Total Params (M)**	**Trainable Params (M)**
Custom CNN	0.62	0.62
ResNet-18 (fine-tuned)	11.18	8.39M
ResNet-18 (full training)	11.18	11.18
ViT-S/16 (fine-tuned)	21.67	7.10

4.1 Custom Convolutional Neural Network

To establish a baseline, we trained a custom Convolutional Neural Network (CNN) on the extracted image tiles of size $(C, 224, 224)$. The model consists of a convolutional feature extractor followed by a compact classification head. Figures 4.1 and 4.2 illustrate the chosen architecture.

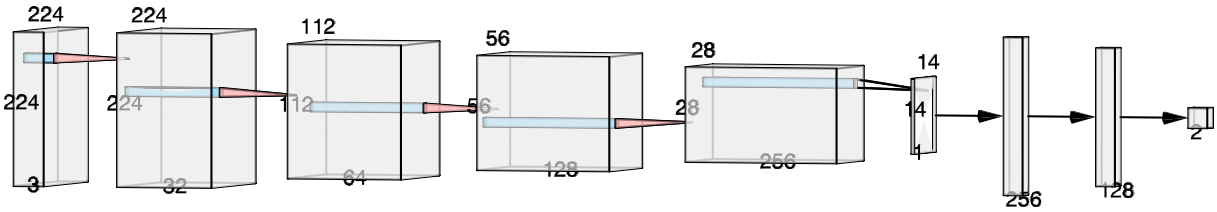


Figure 4.1: Overview of the CNN architecture used in this work. The network processes a 224×224 input tile through four convolutional blocks (see Fig. 4.2 for a detailed illustration of a single block). Each block consists of two 3×3 convolutions (with ReLU activation and Batch Normalization after the first), followed by a 2×2 max-pooling layer that halves the spatial resolution ($224 \rightarrow 112 \rightarrow 56 \rightarrow 28 \rightarrow 14$) while increasing the number of feature channels ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256$). The final $256 \times 14 \times 14$ feature map is reduced to a vector using adaptive average pooling, followed by two fully connected layers ($256 \rightarrow 128 \rightarrow 2$) to produce the output logits. Dropout is applied within each block and before the final classification layer.

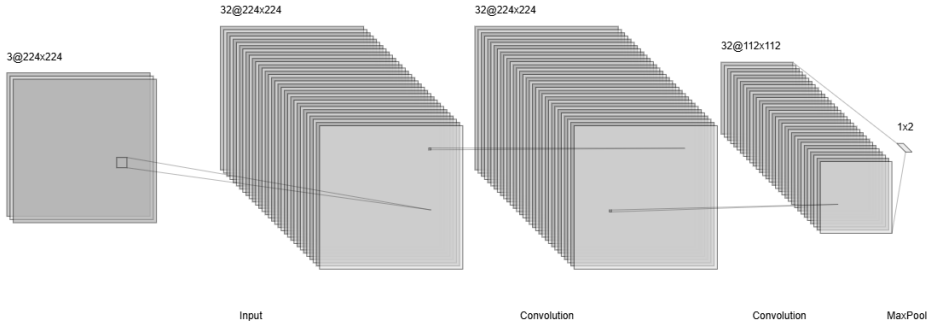


Figure 4.2: Detailed illustration of a single convolutional block from the CNN architecture shown in Fig. 4.1. Each block applies two 3×3 convolutions that preserve spatial resolution, followed by ReLU activations. The first convolution includes Batch Normalization, and Dropout is applied inside the block. A 2×2 max-pooling layer then halves the spatial resolution while keeping the number of feature channels constant. This structure is repeated across all four blocks in the full architecture.

A minimum receptive field (calculation found in D) that just covers the smallest qualifying risk region should be around 50–57 pixels for 224 pixel tiles. In practice, additional headroom is desirable to capture shape, texture, and contextual cues, giving a practical receptive-field target of 60–120 pixels. The chosen architecture's receptive field of roughly 76 pixels sits comfortably within this range. In addition, the CNN preserves spatial detail in the early layers. The first down-sampling operation only occurs after the number of filters has been increased to 32, allowing the model to learn fine-grained, spatially detailed features and a diverse set of local representations before any spatial information is lost. This is beneficial for our UAV data, where blackleg risk-related cues, such as collapsed stems or subtle texture differences are highly local.

4.2 ResNet18

The ResNet architectures were selected due to their strong performance in image classification tasks and their proven ability to generalize across diverse visual domains.

The ResNet18 network consists of 18 learnable layers organized into an initial stem followed by four main stages of residual blocks. The stem includes a 7×7 convolutional layer with stride 2, followed by batch normalisation, a ReLU activation, and a 3×3 max-pooling layer. This component reduces the spatial resolution early on while increasing the channel depth, preparing the feature maps for deeper processing. Each of the four subsequent stages contains two residual blocks. Each residual block is composed of two 3×3 convolutional layers, each followed by batch normalisation and a ReLU activation. A defining characteristic of ResNet architectures is the identity skip connection, which adds the input of the block directly to its output, enabling stable gradient flow during training. To implement the pretrained versions of these models, we use the pretrained ResNet18 weights provided by torchvision[1]. These models are pretrained on the ImageNet-1k dataset.

4.3 Vision Transformer

The ViT-S/16 architecture was selected as a compact Vision Transformer variant well suited for size limited datasets. With an embedding dimension of 384, 12 transformer blocks, and 6 attention heads, ViT-S/16 provides sufficient representational capacity while reducing the risk of overfitting compared to larger ViT variants. The smallest relevant risk region is approximately a 50×50 pixel area (corresponding to a blob with a diameter of about 57 pixels). With a patch size of 16×16 pixels, such a region spans approximately $50/16 \approx 3.1$ patches along each spatial dimension, and is therefore represented by roughly 3×3 tokens in the ViT input sequence. Representing the region of interest with multiple tokens ensures that the relevant visual structure is not collapsed into a single averaged embedding, thereby preserving internal texture, boundaries, and spatial variation that are important for accurate risk detection. For the implementation of ViT-S/16 model we download a pretrained version from huggingface. [2]. The ViT-Small model used in this study is not available on HuggingFace, as Google did not release official weights for this variant. To enable experimentation, we used a HuggingFace-compatible checkpoint converted from the original TIMM implementation

[1] `https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html`

[2] `https://huggingface.co/WinKawaks/vit-small-patch16-224`

Chapter 5

Experiments and Results

This chapter describes what experiments we perform in order to investigate research questions given in section 1.1. First we describe the evaluation metrics which we are going to use to evaluate our experiments.

Table 5.1: Overview of the experiments conducted for blackleg risk detection and their relation to the research questions.

Experiment	**Goal**	**Description**
CNN Baselines	Establish a reproducible baseline and study class imbalance handling	Relates to RQ 1 and RQ 2. We evaluate whether the proposed CNN can separate *Risk* and *No-Risk* tiles and compare multiple class imbalance strategies to determine which approach best supports reliable risk detection.
Pretraining Effect	Investigate whether model complexity and pretraining improve detection performance	Relates to RQ 1, RQ 3, and RQ 4. We compare a ResNet-18 trained from scratch with a partially fine-tuned ImageNet-pretrained ResNet-18 to assess the effect of increased model complexity and transferred representations for UAV-based blackleg risk detection.
Vision Transformer Evaluation	Evaluate the suitability of transformer-based architectures for blackleg risk detection	Relates to RQ 1 and RQ 3. We investigate whether a ViT-S/16 model can outperform convolution-based approaches by leveraging global self-attention, and compare its performance and training behavior against the CNN and ResNet baselines.

5.1 Evaluation

Model performance was evaluated using four standard metrics: accuracy, recall, precision and F_1-score.

5.1.1 Accuracy

Accuracy measures the overall proportion of correctly classified tiles and is defined as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP, TN, FP, and FN denote the numbers of true positives, true negatives, false positives, and false negatives, respectively.

5.1.2 Recall

Recall quantifies the proportion of samples from a given class that are correctly identified:

$$\text{Recall} = \frac{TP}{TP + FN}.$$

While recall can be computed for any class, in this work we place particular emphasis on recall for the *Risk* class, as it reflects the proportion of potentially infected areas that are successfully detected by the model.

5.1.3 Precision

Precision quantifies the proportion of samples predicted as belonging to a given class that are correctly identified:

$$\text{Precision} = \frac{TP}{TP + FP}.$$

In this work we place particular emphasis on precision for the *Risk* class, as it reflects the proportion of predicted risk areas that correspond to truly infected regions and therefore indicates the rate of false alarms produced by the model.

5.1.4 F1-score

The F_1-score represents the harmonic mean between precision and recall:

$$\text{F}_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$

This metric provides a balanced evaluation of both false negatives and false positives. Throughout the experiments, special attention is given to the F1_{risk}. We use it as the objective score for our hyperparameter optimization. This is because this metric directly reflects the model's ability to identify diseased or high-risk regions while keeping false positives low. An outcome that is most relevant to farmers and field inspectors as described in section 1.

5.2 Model training and hyperparameter optimization

All experiments in this thesis follow a common training and hyperparameter optimization procedure. Models are implemented in PyTorch and trained on the dataset described in Section 3.2 using a single NVIDIA RTX 2070 GPU. Hyperparameter optimization is performed using Optuna for all models evaluated in Experiments 1–3. Each optimization run consists of 20 trials, where each trial corresponds to training and evaluating a model with a specific hyperparameter configuration. Based on previously observed objective values, Optuna proposes new configurations for subsequent trials. To prevent spatial leakage between training and validation data, tiles are grouped by acquisition block and evaluated using StratifiedGroupKFold. This ensures that all tiles from the same acquisition block are assigned to the same fold while maintaining approximately stratified class distributions. Each trial is trained for a maximum of 40 epochs using per-fold early stopping with a patience of five epochs and a minimum improvement threshold of $\Delta_{\text{min}} = 10^{-4}$. Early stopping is primarily monitored using the optimization objective (F1_{risk}) on the validation set. As this metric can be unstable during training, validation loss is used as a secondary criterion. Training is terminated only if no improvement is observed for five consecutive epochs in either metric. Unless otherwise stated, hyperparameter optimization and model selection are performed exclusively on the training and validation data. Held-out test blocks are never used during optimization and are reserved solely for final evaluation.

5.3 Experiment 1: Baseline CNN Training

The purpose of this experiment is to establish a baseline for detecting *blackleg risk areas* from UAV imagery using a custom CNN architecture (see Section 4.1). The experiment evaluates whether the proposed CNN can successfully separate *Risk* and *No-Risk* tiles under realistic field conditions. In addition, this experiment investigates the impact of different class imbalance handling strategies on performance, with the goal of identifying approaches that maximize detection of the minority *Risk* class.

5.3.1 Search Space

The hyperparameter search space consists of four class-imbalance strategies described in Section 3.4.1: (i) Random Sampling CE loss - baseline, (ii) Random Sampling CE loss + label smoothing, (iii) Weighted Random Sampling CE, and (iv) Random focal loss. In addition, each trial samples a set of standard deep learning hyperparameters: learning rate ($[10^{-5}, 5 \times 10^{-4}]$, log-uniform), dropout probability ($[0, 0.6]$), weight decay ($[10^{-6}, 10^{-2}]$, log-uniform), optimizer (`Adam`, `AdamW`, or `SGD`), batch size (8, 16, or 32), and data augmentation (*none* or *basic*). Learning rate and weight decay are sampled on a

logarithmic scale to reflect their multiplicative effect and to efficiently cover several orders of magnitude. When SGD is selected, it is used with Nesterov momentum (momentum = 0.9), while Adam and AdamW use default PyTorch settings.

5.3.2 Results

Class Imbalance Strategies and Hyperparameter Optimization

To assess the effect of different class imbalance strategies independently of individual hyperparameter choices, each strategy is evaluated across multiple Optuna trials, where all remaining hyperparameters are jointly optimized. Performance is then summarized based on the distribution of validation scores across trials. Before analyzing the overall performance, we first examine the impact of the different class imbalance strategies. As shown in Figure 5.1, weighted random sampling with cross-entropy loss yields the strongest overall performance, indicating that increased sampling of the minority (*Risk*) class during training is beneficial for this task.

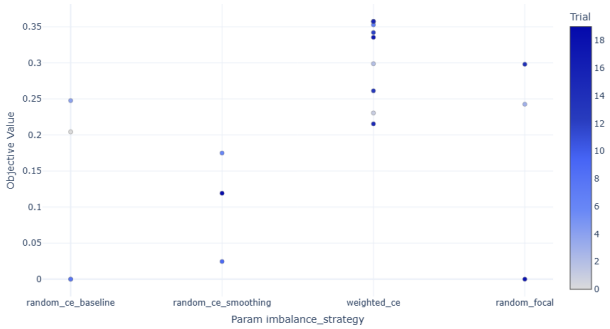


Figure 5.1: Objective values ($F1_{risk}$) across class imbalance strategies in the CNN Optuna search.

Table 5.2: Optimal hyperparameter configuration selected by Optuna for the baseline CNN.

Strategy	**Batch**	**Weight Decay**	**Optimizer**	**LR**	**Dropout**	**Aug.**
iii	32	3×10^{-4}	Adam	6.98×10^{-5}	0.09	Basic

Table 5.2 reports the hyperparameter configuration selected by Optuna for the baseline CNN. The selected model uses Weighted Random Sampling CE loss (iii) with a batch size of 32, weight decay of 3×10^{-4}, the Adam optimizer, a learning rate of 6.98×10^{-5}, a dropout probability of 0.09, and basic data augmentation.

Performance

Table 5.3: Cross-validation and test performance of the selected baseline CNN model.

Split	Accuracy	$\text{Recall}_{\text{risk}}$	$\text{Precision}_{\text{risk}}$	F1_{risk}
Cross-validation	0.924	0.519	0.270	0.357
Test (held-out)	0.855	0.483	0.438	0.460

Under this configuration, the baseline CNN achieves a mean cross-validation F1_{risk} score of 0.357 with a validation accuracy of 0.924. When evaluated on the held-out test blocks (Table 5.3), the model attains a higher F1_{risk} score of 0.460, primarily driven by an increase in $\text{Precision}_{\text{risk}}$. We note that the test set contains a higher proportion of *Risk* tiles than the validation folds.

5.4 Experiment 2: ResNet and Pretraining Effects

The goal of Experiment 2 is to assess whether increased model complexity and ImageNet pretraining improve the detection of blackleg risk areas. This experiment relates to Research Questions RQ1, RQ3, and RQ4. We evaluate a ResNet-18 architecture under two training regimes: (i) full end-to-end training from scratch and (ii) initialization with ImageNet-1K pretrained weights followed by partial fine-tuning. In the latter case, only the final residual stage is fine-tuned, while earlier layers remain frozen. This design allows the model to adapt high-level representations to the target UAV domain while preserving generic pretrained features, and limits the number of trainable parameters to 8.39 M (Table 4.1).

5.4.1 Transfer Learning Setup

The training and evaluation protocol from Experiment 1 is reused. To reduce the number of free hyperparameters, the class imbalance handling strategy is fixed to the best-performing approach from Experiment 1, and the hyperparameter search space is restricted to the parameters described in Section 5.4.2. During fine-tuning, the final residual block and the classification head are unfrozen, while all earlier layers remain frozen. Discriminative learning rates are employed, with a higher learning rate assigned to the classification head than to the fine-tuned residual stage, reflecting the need for stronger adaptation in task-specific layers. For consistency with the ImageNet-pretrained ResNet-18 backbone, all input images are normalized using ImageNet mean and standard deviation values.[1]

[1] `https://docs.pytorch.org/vision/stable/transforms.html`

5.4.2 Search Space

For each trial, we sample learning rates for the classification head ($[10^{-5}, 5 \times 10^{-4}]$, log-uniform) and the last residual block ($[10^{-6}, 10^{-4}]$, log-uniform), along with dropout probability ([0, 0.6], applied to the classification head only) and weight decay ($[10^{-6}, 10^{-2}]$, log-uniform). In addition, we perform categorical searches over the optimizer (`Adam`, `AdamW`, `SGD`), batch size (8, 16, or 32), and data augmentation strategy (*none* or *basic*).

5.4.3 Results

Table 5.4 summarizes the optimal hyperparameter configurations identified by the Optuna search for the two ResNet-18 training regimes. For the fine-tuned setup, Optuna selects the Adam optimizer and a dropout rate of 0.289, whereas full end-to-end training favors SGD combined with a higher dropout rate of 0.579. Batch size and data augmentation are identical across both regimes (batch size 8 with basic augmentation) and are therefore omitted from the table.

Table 5.4: Optimal hyperparameter configurations selected by Optuna for ResNet-18 under different training regimes. Batch size and data augmentation are fixed across both regimes and therefore omitted. For full end-to-end training, a single learning rate is used and the LR (Blocks) parameter is not applicable.

Training	**Weight Decay**	**Optimizer**	**LR (Head)**	**LR (Blocks)**	**Dropout**
Fine-tuned	6.07×10^{-3}	Adam	2.40×10^{-4}	9.19×10^{-5}	0.289
Full training	5.46×10^{-4}	SGD	4.09×10^{-4}	–	0.579

Performance

As shown in Table 5.5, the fine-tuned model's accuracy decreases from 0.939 in cross-validation to 0.778 on the test set, while full end-to-end training shows a smaller reduction from 0.896 to 0.867. The fine-tuned model achieves a higher test $\mathrm{F1}_{\mathrm{risk}}$ (0.423 vs. 0.386), driven primarily by a substantially higher $\mathrm{Recall}_{\mathrm{risk}}$ (0.639 vs. 0.326), at the cost of lower $\mathrm{Precision}_{\mathrm{risk}}$. In contrast, full end-to-end training yields more conservative predictions with higher precision but misses a larger fraction of risk tiles.

Table 5.5: Cross-validation and test performance of ResNet-18 under different training regimes. Cross-validation results correspond to the Optuna-selected models optimized for $F1_{risk}$, while test results are evaluated on held-out test blocks.

Training Mode	**Split**	**Acc**	**Recall$_{risk}$**	**Precision$_{risk}$**	**F1$_{risk}$**
Fine-tuned (Last 2 blocks)	Cross-val	0.939	0.464	0.313	0.374
Fine-tuned (Last 2 blocks)	Test	0.778	0.639	0.316	0.423
Full end-to-end training	Cross-val	0.896	0.520	0.196	0.285
Full end-to-end training	Test	0.867	0.326	0.470	0.386

5.5 Experiment 3: Vision Transformer Evaluation

The goal of Experiment 3 is to investigate whether a transformer-based architecture can outperform the convolution-based models evaluated in previous experiments for blackleg risk detection. In this experiment, we fine-tune an ImageNet-pretrained ViT-S/16 model, as training a Vision Transformer from scratch would likely lead to overfitting given the limited dataset size. Fine-tuning is restricted to the final four transformer blocks, corresponding to approximately 7.1 M of the 21.7 M total model parameters (Table 4.1).

5.5.1 Vision Transformer search space

For the ViT-S/16 experiments, the hyperparameter search space is adapted to reflect standard fine-tuning practices for transformer-based architectures. Separate learning rates are optimized for the classification head and the transformer backbone. The head learning rate is sampled log-uniformly from $[1 \times 10^{-5}, 3 \times 10^{-4}]$, while a more conservative backbone learning rate is sampled log-uniformly from $[3 \times 10^{-6}, 5 \times 10^{-5}]$ to account for the higher sensitivity of pretrained transformer weights.

The dropout probability is sampled uniformly from the interval [0, 0.4] and applied to the classification head. The batch size is selected from $\{8, 16\}$. Weight decay is sampled log-uniformly from $[1 \times 10^{-5}, 1 \times 10^{-2}]$. The optimizer is chosen from `Adam`, `AdamW`, or `SGD`, and the data augmentation strategy is selected as either *none* or *basic*.

5.5.2 Results

Table 5.6 reports the optimal hyperparameter configuration selected by the Optuna search for the ViT-S/16 model. The Optuna search selected a batch size of 8 and basic data augmentation; these parameters are omitted from the table for compactness.

Table 5.6: Optimal hyperparameter configuration selected by Optuna for the ViT-S/16 model.

Model	Weight Decay	Optimizer	LR (Head)	LR (Blocks)	Dropout
ViT-S/16	9.46×10^{-5}	AdamW	7.64×10^{-5}	4.01×10^{-5}	0.035

Performance

Table 5.7: Cross-validation and test performance of the ViT-S/16 model. Cross-validation results correspond to the Optuna-selected configuration optimized for $F1_{risk}$, while test results are evaluated on held-out test blocks.

Model	Split	M. Acc	M. $Recall_{risk}$	M. $Precision_{risk}$	M. $F1_{risk}$
ViT-S/16	Cross-val	0.945	0.410	0.347	0.376
ViT-S/16	Test	0.837	0.544	0.398	0.460

As shown in Table 5.7, the ViT-S/16 model achieves improved performance on the held-out test blocks compared to cross-validation, with an increase in $F1_{risk}$ from 0.376 to 0.460 and $Recall_{risk}$ from 0.410 to 0.544, despite a decrease in overall accuracy. The resulting test performance is comparable to that of the baseline CNN.

5.6 Overall Model Performance on Held-Out Test Data

Table 5.8: Test performance comparison across all evaluated models on held-out test blocks.

Model	Accuracy	$Recall_{risk}$	$Precision_{risk}$	$F1_{risk}$
Custom CNN	0.855	0.483	0.438	**0.460**
ResNet-18 (fine-tuned)	0.778	**0.639**	0.316	0.423
ResNet-18 (full training)	**0.867**	0.326	**0.470**	0.386
ViT-S/16	0.837	0.544	0.398	**0.460**

Table 5.8 compares the test performance of all evaluated model architectures. Overall accuracy is comparable across models, with the exception of the fine-tuned ResNet-18, which attains a lower accuracy of 0.778. The fine-tuned ResNet-18 achieves the highest $Recall_{risk}$ by a clear margin.

Notably, the two models initialized with ImageNet pretraining (the fine-tuned ResNet-18 and ViT-S/16) achieve higher $Recall_{risk}$ but lower $Precision_{risk}$ than the models trained from scratch, indicating a consistent recall–precision trade-off associated with pretraining in this setting.

The Custom CNN and ViT-S/16 models obtain the highest $F1_{risk}$ scores, indicating the best overall balance between precision and recall. Between these two models, ViT-S/16 exhibits higher $Recall_{risk}$ but lower $Precision_{risk}$ compared to the Custom CNN.

5.7 Tile-Level Prediction Examples

This section describes a few samples of tile-level predictions from the Custom CNN model. Figure 5.2 presents four representative false negative predictions from the test set, along with the corresponding predicted risk probabilities, which indicate the model's confidence in each classification. For the first three examples, the model assigns a low probability to the *Risk* class, indicating high confidence in a *No-Risk* prediction. In contrast, the final example (index 80) exhibits a higher predicted risk probability, suggesting a more uncertain classification.

Figure 5.2: Examples of **false negative** predictions on the test set. Each tile corresponds to a ground-truth *Risk* area that was incorrectly classified as *No-Risk*. The displayed probability indicates the model's predicted confidence for the Risk class.

Figure 5.3 shows representative examples of true positive predictions, where blackleg risk areas are correctly identified by the model. For the examples with index 210 and 211, the model assigns a high predicted risk probability, indicating strong confidence in the classification. In contrast, the example with index 212 exhibits a lower predicted risk probability, reflecting a more uncertain prediction.

Figure 5.3: Examples of **true positive** predictions on the test set. Each tile shows a correctly identified *Risk* area, along with the model's predicted probability for the Risk class.

Chapter 6

Discussion and Conclusion

6.1 Overview of Findings

This chapter discusses the experimental results in relation to the four research questions presented in Section 1.1. Overall, the findings demonstrate that machine learning models can, in some instances, detect blackleg risk areas from UAV imagery. In Experiment 1, class imbalance handling had a clear impact on performance. Strategies that explicitly accounted for the skewed class distribution outperformed naive approaches, highlighting class imbalance as a central challenge for blackleg risk detection. With respect to model architectures, the Custom CNN and the Vision Transformer achieved the strongest overall performance, exhibiting comparable $F1_{risk}$ scores on the held-out test data. While predictive performance was similar, the Custom CNN requires significantly fewer computational resources and achieves slightly higher accuracy, making it more suitable for practical deployment. Pretrained models generally achieved higher $Recall_{risk}$ and $F1_{risk}$ values, whereas the model trained from scratch exhibited higher precision and accuracy, indicating a trade-off between sensitivity and conservative predictions.

6.2 Research Question 1: Detectability of Blackleg Risk Areas

Research Question 1 examines whether it is possible to train a model capable of detecting *blackleg risk areas* from UAV imagery. At a minimum, this requires that the model identifies risk tiles in the held-out test set. Qualitative examples of true positive predictions (Figure 5.3)demonstrate that the models successfully detect several expert-annotated risk areas. Some examples exhibit clear visual cues, such as collapsed plants, which are consistently associated with risk predictions. Other true positive tiles show less visually obvious symptoms and are classified with lower predicted risk probabilities, indicating that the models do not rely solely on easily identifiable visual patterns. While informative,

these qualitative examples allow only limited reasoning about overall model behavior and do not provide a comprehensive explanation of the learned decision criteria. Quantitatively, as shown in Table 5.8, both the Custom CNN and the Vision Transformer achieve a test $F1_{risk}$ score of approximately 0.46. This demonstrates that the models are able to identify a meaningful subset of risk tiles in the held-out test data without collapsing into a trivial all-risk prediction strategy. Together, these qualitative and quantitative results confirm that blackleg risk areas are, to a non-trivial extent, detectable from UAV imagery using the evaluated models.

6.3 Research Question 2: Handling Class Imbalance

Figure 5.1 shows that the weighted random sampler combined with cross-entropy loss achieves the highest objective values during hyperparameter optimization, indicating that this approach is the most effective among the evaluated class imbalance strategies. A likely explanation is that the weighted sampler directly compensates for the skewed class distribution by increasing the representation of the minority class, while retaining the stability of a standard loss function. In contrast, less invasive approaches such as random sampling with label smoothing may provide insufficient correction, whereas Focal Loss depends on multiple interacting sub-parameters (e.g., α and γ), making it more difficult to optimize reliably in limited-data hyperparameter searches. For this dataset, sampling-based class imbalance handling using a weighted sampler combined with cross-entropy loss consistently yields higher and more stable $F1_{risk}$ scores than loss-based alternatives

6.4 Research Question 3: Model Architecture Comparison

The results reveal clear trade-offs between model architectures rather than a single dominant solution. While the Custom CNN and the Vision Transformer achieve the highest $F1_{risk}$ scores, the ResNet-18 variants exhibit complementary strengths, such as higher recall or higher precision depending on the training regime. Although it is hard to attribute these behaviors to single factors, several plausible explanations can be proposed.

One explanation for the lower $F1_{risk}$ scores observed for the ResNet-18 models may relate to their architectural design, in particular the aggressive early downsampling performed by the initial convolution and pooling layers. While this design is computationally efficient and well-suited for ImageNet-style classification, it may be less optimal for blackleg risk detection, where relevant cues can consist of fine-grained local spatial details. In contrast, the Vision Transformer (ViT-S/16), which achieves $F1_{risk}$ performance comparable to the Custom CNN, integrates global spatial context through self-attention from the earliest layers. This allows spatial relationships across the entire tile to be modeled directly, without

relying on early spatial downsampling. As a result, the Vision Transformer may better preserve subtle spatial patterns relevant to risk detection, helping to explain its stronger balanced performance. The fact that the Custom CNN achieves performance comparable to, or better than, ResNet-18 also suggests that increasing architectural complexity is not always necessary for this task. While many recent studies emphasize deeper or more complex architectures, our results indicate that a lightweight model tailored to the specific characteristics of blackleg risk detection can achieve competitive performance. Taken together, these observations highlight clear trade-offs between model choices. Depending on stakeholder priorities, farmers may prefer a model with strong recall at the expense of precision, such as the fine-tuned ResNet-18, or a more balanced solution like the Custom CNN, which also offers lower computational cost and easier deployment.

6.5 Research Question 4: Pretraining Effect

When comparing the two ResNet-18 models evaluated in Experiment 2, fine-tuning from ImageNet-pretrained weights results in an improved $F1_{risk}$ score compared to training from scratch, indicating that pretraining has a positive effect in this setting. Furthermore, when comparing the two pretrained models—the fine-tuned ResNet-18 and the ViT-S/16—it can be observed in Table 5.8 that both achieve higher $Recall_{risk}$ than the non-pretrained models, while exhibiting lower $Precision_{risk}$.

Although this observation does not establish a causal relationship, it suggests that ImageNet pretraining may improve sensitivity to minority-class risk patterns. A plausible explanation is that pretraining provides more robust low- and mid-level feature representations, which facilitate adaptation to subtle visual cues associated with blackleg risk in a limited-data setting. Taken together, these results indicate that ImageNet pretraining is associated with higher recall for the minority risk class on this dataset, although this improvement does not translate into consistently superior performance across all evaluation metrics.

6.6 Limitations

6.6.1 Dataset Limitations

Image Quality

All imagery used in this study was captured using UAVs, meaning that flight height, camera quality, and lighting conditions directly influence the effective resolution and clarity of the images. In our case, the ground sampling distance was insufficient to reliably observe clear blackleg symptoms such as stem rot, and make sure if they actually stemed from blackleg and not another reason. As a result, the task had to be re-framed from detecting individual

diseased plants to identifying broader *blackleg risk areas*. This re-framing also introduced additional difficulty for the annotators, who reported that the resolution often made it challenging to distinguish between genuine disease-related signals and natural variation in crop structure. For future data collection, it would be valuable to evaluate which flight heights provide a ground sampling distance that is high enough for symptom-level visibility while still allowing large fields to be surveyed efficiently.

Annotations

Detecting blackleg risk areas is inherently uncertain, as the definition of risk involves subjective judgment based on subtle visual cues. To better assess annotation uncertainty and reduce potential single-annotator bias, a second annotator was recruited. The initial intention was to use the intersection of the two annotation sets as a conservative ground truth. However, the intersection between the two risk annotation sets was approximately 13%, indicating a low level of agreement. Using only the intersection would therefore resulting in very few risk samples and further worsening the existing class imbalance. It is worth noting that there was an experience gap of approximately 15 years between the two annotators, which may have influenced their respective interpretations of risk. The observed disagreement does not necessarily imply that either annotator was incorrect; rather, it highlights the difficulty of the task and the ambiguity of the visual cues available at the given spatial resolution. This level of annotation disagreement has several important implications. First, it introduces uncertainty into the training labels, which may negatively affect model learning and bias the model toward annotator-specific interpretations of risk. As a result, reported performance metrics should be interpreted with caution, as the model is trained under imperfect supervision. Second, some predictions that appear incorrect when compared to the reference annotations may in fact reflect genuine risk-related ambiguity in the underlying imagery rather than true model errors. Overall, the limited agreement between annotators emphasizes that the ground-truth labels are inherently noisy. Annotation uncertainty is therefore likely to contribute to both false positives and false negatives, and the reported results should be viewed as conservative estimates of model performance under uncertain supervision.

Image Distortion

During orthophoto generation, certain regions, particularly near image boundaries, were subject to geometric distortion. These distortions can produce visual patterns that do not correspond to true field conditions and may confuse machine learning models during training and evaluation. By altering local texture and geometry, such effects can impact both annotation quality and model predictions. An example of this effect is shown in Figure 5.2 (index 77). Although unavoidable to some extent in UAV-based orthomosaic construction, these distortions constitute an additional source of noise in the dataset and

may negatively affect model performance.

6.6.2 Methodological Limitations

Spatial Leakage

When orthomosaics are split into tiles and these tiles are grouped into spatial blocks for training, validation, and testing, some degree of *spatial leakage* is unavoidable. Although the dropped margin shown in Figure 3.2 prevents most direct spatial overlap, neighboring blocks may still share similar illumination conditions, soil appearance, crop density, or other local characteristics. As a result, tiles belonging to different data splits can remain partially correlated, potentially leading to slightly optimistic estimates of generalization performance. One way to further mitigate spatial leakage would be to introduce buffer regions between the training, validation, and test sets by discarding all blocks that lie directly between partitions. However, this approach comes at a substantial cost, as it removes a significant portion of the data. This is particularly problematic in our case, where *Risk* tiles constitute only about 5% of the dataset. Further reducing the dataset could meaningfully weaken the model's ability to learn a stable decision boundary. A more targeted compromise would be to discard only *No-Risk* tiles within the buffer blocks while retaining the rarer *Risk* tiles. This would reduce spatial leakage primarily for the majority class while preserving the minority-class signal. In this work, we opted for the simpler and more data-efficient strategy of retaining all blocks, while explicitly accounting for the potential effects of spatial leakage when interpreting the results.

6.6.3 Model limitations

Visual inspection of misclassified tiles in 5.2 reveals that the model struggles primarily in cases where blackleg risk cues are subtle or ambiguous at the available image resolution. *Risk* tiles misclassified as *No-Risk* show no clear visual symptoms such as collapsed plants, leading to confident but incorrect predictions. In contrast, many correctly classified *Risk* tiles contain exposed soil, suggesting that the model relies on coarse visual cues associated with plant collapse rather than disease-specific symptoms. These cues are not consistently present and can also result from non-disease-related disturbances, which may limit the model's ability to generalize robustly. Finally, these qualitative observations provide only limited insight into the model's decision process. Without explicit explainability methods, it remains unclear which image regions drive predictions, making it difficult to determine whether the model relies on meaningful disease cues or spurious correlations.

6.6.4 Practical Limitations

The practical deployment of the model is constrained by several factors related to the data and the biological variability of the crop. First, all imagery used for training and evaluation was collected on a single day. As a result, the model may perform worse on images captured under different illumination, weather, or contrast conditions. Similarly, the dataset represents only one growth stage of the potato crop. Changes in canopy structure and plant size at different growth stages may alter the visual cues the model relies on, limiting its ability to generalize beyond the conditions seen during training. In addition, the imagery contains only a single potato variety. Different varieties can exhibit distinct growth habits, canopy density, and leaf morphology, which may influence the appearance of both healthy and stressed plants. Without additional training data, the model may therefore struggle to generalize to fields containing other varieties. For these reasons, the current model should primarily be viewed as a proof of concept demonstrating that blackleg risk areas can be detected from RGB UAV imagery. If similar performance were achieved on an unseen field, the model would detect approximately half of the risk areas while producing some false positives. In practice, this level of performance is not sufficient for automated decision-making, but it may still be useful as a high-level decision-support tool to help prioritize where manual inspection efforts should be focused.

6.7 Future Work

6.7.1 More Data and Improved Annotations

A primary direction for future work is the collection of higher-quality and more diverse data. During annotation, certain regions of the field were deemed unsuitable for risk annotation due to disturbances such as tractor tracks, which obscured the canopy. Combined with the artifact discussed in section 6.6.1 Future datasets should therefore exclude regions with artifacts or sprayer tracks preprocessing to ensure more consistent training data.

More importantly, collecting imagery at higher spatial resolution would substantially improve both annotation quality and model performance. Many visual cues relevant to blackleg risk, such as small canopy gaps, weak plants, or early signs of collapse, are difficult to discern at the current ground sampling distance. Higher-resolution imagery would enable more confident annotations and provide clearer signals for learning.

Future studies should also include data from multiple fields, growth stages, and environmental conditions, as well as involve a larger pool of expert annotators. This would allow for improved label consistency, the development of clearer annotation protocols, and potentially consensus-based ground truth.

Finally, the modeling pipeline developed in this thesis is designed to be reusable and scalable. As additional and improved datasets become available, the same framework can

be retrained and extended, offering a clear path toward more robust and practically useful blackleg risk detection models.

6.7.2 Tile Size

Another interesting parameter to investigate is the tile size. Increasing the tile size allows the model to observe a larger spatial context, potentially capturing patterns that span multiple plants rather than relying solely on local features. This may be particularly beneficial for models architectures such as the Vision Transformer, which leverages global self-attention and can make use of long-range dependencies at early stages of the network. Larger tiles may also better reflect how field professionals work during inspections. Human inspectors rarely evaluate plants in isolation; instead, they look for anomalies in the local neighborhood, such as a collapsed plant surrounded by healthy ones, or inconsistent canopy structure across a patch. By increasing the tile size, the model is given the opportunity to follow a similar intuition and reason over broader spatial cues.

Future experiments could therefore compare model performance across multiple tile sizes to determine whether access to larger spatial context improves risk detection, particularly for models designed to exploit global information.

6.7.3 Classification Map

A natural next step toward practical deployment is to generate a visual *classification map*. In this approach, each predicted tile is mapped back onto the original orthophoto, preserving its spatial location and scale. The resulting map provides an intuitive overlay showing where the model detects potential risk areas. By linking each tile to its corresponding geographic coordinates, inspectors or farmers can easily identify the exact GPS position of a suspected risk zone. This makes the predictions actionable in a real-world setting, as it allows field teams to focus their time and attention on the areas most likely to contain problems. In addition to showing where the model predicts risk, it would also be valuable to visualize how confident the model is in its predictions. This could be done by creating a heatmap of uncertainty or confidence scores, where tile colors gradually shift based on the model's probability estimates. Areas with high uncertainty might indicate borderline cases, noisy annotations, or parts of the field where visual cues are weak or ambiguous. Such a heatmap would help inspectors prioritize their efforts even further: high-confidence detections can be checked quickly, while high-uncertainty regions may require a closer look or additional imaging. It would also make the model's behavior more transparent and easier to interpret, which is especially important in agricultural decision-making where trust and explainability matter.

6.7.4 Polygon-Based Metric

Future work could explore polygon-based evaluation metrics that better reflect the spatial structure of the annotations. As illustrated in Figure 5.2, the model fails to detect risk in tiles with indices 77–80. The fact that these tiles are consecutive is consistent with them overlapping the same risk annotation polygon, although this cannot be confirmed without explicit polygon-level analysis. In contrast, the model may detect risk in a neighboring tile (index 81), suggesting that partial detection of a risk region can occur even when several tiles are misclassified. In such cases, tile-level metrics may over-penalize the model. A potentially more informative alternative would therefore be to evaluate the proportion of annotated risk polygons that are detected at least once by the model, rather than counting the number of individual tiles classified as Risk.

6.7.5 Model Improvement

Future work could further investigate the hypothesis discussed in Section 6.4 that the performance differences between the Custom CNN and ResNet-18 are partly driven by early downsampling in the ResNet architecture. One way to test this hypothesis would be to modify ResNet-18 by reducing or removing its initial downsampling operations, for example by removing the max-pooling layer or changing early stride-2 convolutions to stride-1, and then comparing the resulting model to the original ResNet-18 and the Custom CNN. An alternative experiment would be to introduce early downsampling into the Custom CNN and evaluate whether this leads to a degradation in $F1_{\text{risk}}$ performance. Together, such controlled architectural modifications could help clarify which architectural properties are most critical for effective blackleg risk detection.

In addition, future work could further investigate the role of pretraining by exploring different transfer learning strategies, such as varying the number of fine-tuned layers or selectively freezing parts of the network. This would help determine how much task-specific adaptation is required to benefit from ImageNet pretraining in this context.

Chapter 7

Conclusion

This thesis investigated whether blackleg risk areas in industrial seed potato fields can be detected from RGB imagery collected using a UAV. A complete machine learning pipeline was developed, encompassing data preprocessing, expert-driven annotation, model training, and evaluation under conditions of severe class imbalance.

The results demonstrate that blackleg risk areas can be detected to a non-trivial extent, with the best-performing models achieving an $F1_{risk}$ score of approximately 0.46 on held-out spatial test data. For the dataset and experimental setup considered in this thesis, sampling-based class imbalance handling using a weighted sampler proved most effective. No single model architecture dominated across all evaluation metrics; instead, clear trade-offs were observed between recall, precision, accuracy, and computational cost. The custom CNN achieved the most balanced overall performance, while ImageNet pretraining was associated primarily with higher recall for the minority risk class, typically at the cost of lower precision. Overall, this work provides a proof of concept that UAV-based RGB imagery can support blackleg risk detection under realistic industrial field conditions. At the same time, performance is constrained by data quality, annotation uncertainty, and limited spatial. With improved annotations and more diverse datasets, the proposed framework can be reused and extended toward the development of a practical decision-support tool to support targeted field inspections in seed potato production.

Chapter 8

Code and data

The code for the project can be found at `https://github.com/emil5654/Master-Thesis-public`
The data is available at request to au668867@uni.au.dk

List of Abbreviations

BEiT Bidirectional Encoder representation from Image Transformers

CE Cross-Entropy

CNN Convolutional Neural Network

ResNet Residual Neural Network

RGB Red–Green–Blue

RQ Research Question

SVM Support Vector Machine

TPE Tree-Structured Parzen Estimator

UAV Unmanned Aerial Vehicle

ViT Vision Transformer

Bibliography

[1] Danmarks Statistik. *Areal med læggekartofler i Danmark (2004–2024)*. 2024. URL: https://www.dst.dk/da/Statistik/nyheder-analyser-publ/nyt/NytHtml?cid=48805 (visited on 08/20/2024).

[2] Landbrugsstyrelsen. *Vejledning om læggekartofler og avl af konsumkartofler*. Technical guideline, Landbrugsstyrelsen, Danmark. Aug. 2024. URL: https://lbst.dk/Media/638603391155405705/Aug%202024%20-%20Vejledning%20om%20l%C3%A6ggekartofler%20og%20avl%20af%20konsumkartofler%20.pdf.

[3] Rafał Czajkowski, Wilis de Boer, Jacques A. van Veen, and Jan M. van der Wolf. "Ecology and control of *Dickeya* spp. in potato". In: *Plant Pathology* 60.7 (2011), pp. 999–1013. DOI: 10.1111/j.1365-3059.2011.02588.x. URL: https://scholarlypublications.universiteitleiden.nl/access/item%3A2891398/download.

[4] G.W. Hulsman, J. Heijden, and Ricardo Torres. "Automated Detection and Localization of Potato Blackleg Using a Convolutional Neural Network and Activation Maps". In: *Potato Research* 68 (2025). DOI: 10.1007/s11540-025-09873-x.

[5] Tajul Miftahushudur, Halil Mertkan Sahin, Bruce Grieve, and Hujun Yin. "A Survey of Methods for Addressing Imbalance Data Problems in Agriculture Applications". In: *Remote Sensing* 17.3 (2025). ISSN: 2072-4292. DOI: 10.3390/rs17030454. URL: https://www.mdpi.com/2072-4292/17/3/454.

[6] Manya Afonso, Pieter Blok, Gerrit Polder, Jan Wolf, and Jan Kamp. "Blackleg Detection in Potato Plants using Convolutional Neural Networks". In: *IFAC-PapersOnLine* 52 (Jan. 2019), pp. 6–11. DOI: 10.1016/j.ifacol.2019.12.481.

[7] Marston Héracles Domingues Franceschini, Harm Bartholomeus, Dirk Frederik van Apeldoorn, Juha Suomalainen, and Lammert Kooistra. "Feasibility of Unmanned Aerial Vehicle Optical Imagery for Early Detection and Severity Assessment of Late Blight in Potato". In: *Remote Sensing* 11.3 (2019). DOI: 10.3390/rs11030224. URL: https://www.mdpi.com/2072-4292/11/3/224.

[8] Gerrit Polder, Pieter M. Blok, Hendrik A. C. de Villiers, Jan M. van der Wolf, and Jan Kamp. "Potato Virus Y Detection in Seed Potatoes Using Deep Learning on Hyperspectral Images". In: *Frontiers in Plant Science* Volume 10 - 2019 (2019). ISSN: 1664-462X. DOI: 10.3389/fpls.2019.00209. URL: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2019.00209.

[9] Julio M. Duarte-Carvajalino, Diego F. Alzate, Andrés A. Ramirez, Juan D. Santa-Sepulveda, Alexandra E. Fajardo-Rojas, and Mauricio Soto-Suárez. "Evaluating Late Blight Severity in Potato Crops Using Unmanned Aerial Vehicles and Machine Learning Algorithms". In: *Remote Sensing* 10.10 (2018). DOI: 10.3390/rs10101513. URL: https://www.mdpi.com/2072-4292/10/10/1513.

[10] Sandika Biswas, Bhushan Jagyasi, Bir Pal Singh, and Mehi Lal. "Severity identification of Potato Late Blight disease from crop images captured under uncontrolled environment". In: *2014 IEEE Canada International Humanitarian Technology Conference - (IHTC)*. 2014, pp. 1–5. DOI: 10.1109/IHTC.2014.7147519.

[11] Vincent Espitalier, Jean-Christophe Lombardo, Hervé Goëau, Christophe Botella, Toke Thomas Høye, Mads Dyrmann, Pierre Bonnet, and Alexis Joly. "Adapting a global plant identification model to detect invasive alien plant species in high-resolution road side images". In: *Ecological Informatics* 89 (2025), p. 103129. DOI: 10.1016/j.ecoinf.2025.103129. URL: https://www.sciencedirect.com/science/article/pii/S1574954125001384.

[12] Sana Parez, Naqqash Dilshad, Norah Saleh Alghamdi, Turki M. Alanazi, and Jong Weon Lee. "Visual Intelligence in Precision Agriculture: Exploring Plant Disease Detection via Efficient Vision Transformers". In: *Sensors* 23.15 (2023). DOI: 10.3390/s23156949. URL: https://www.mdpi.com/1424-8220/23/15/6949.

[13] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791. URL: http://vision.stanford.edu/papers/Lecun98.pdf.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2015, pp. 770–778.

[15] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. "How transferable are features in deep neural networks?" In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *An Image is Worth 16x16 Words:*

Transformers for Image Recognition at Scale. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[17] Hangbo Bao, Li Dong, and Furu Wei. "BEiT: BERT Pre-Training of Image Transformers". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2022. URL: https://openreview.net/forum?id=p-BhZSz59o4.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. *Focal Loss for Dense Object Detection*. 2018. arXiv: 1708.02002 [cs.CV]. URL: https://arxiv.org/abs/1708.02002.

[19] Alessandro Bria, Claudio Marrocco, and Francesco Tortorella. "Addressing class imbalance in deep learning for small lesion detection on medical images". In: *Computers in Biology and Medicine* 120 (2020), p. 103735. ISSN: 0010-4825. DOI: https://doi.org/10.1016/j.compbiomed.2020.103735. URL: https://www.sciencedirect.com/science/article/pii/S0010482520301177.

[20] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. DOI: 10.1145/3620665.3640366. URL: https://docs.pytorch.org/assets/pytorch2-2.pdf.

[21] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. "Optuna: A next-generation hyperparameter optimization framework". In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[22] Hilman F. Pardede, Endang Suryawati, Rika Sustika, and Vicky Zilvan. "Unsupervised Convolutional Autoencoder-Based Feature Learning for Automatic Detection of Plant Diseases". In: *2018 International Conference on Computer, Control, Informatics and its Applications (IC3INA)*. 2018, pp. 158–162. DOI: 10.1109/IC3INA.2018.8629518.

[23] emaRex. *PlantVillage Dataset (PlantDisease) on Kaggle*. https://www.kaggle.com/datasets/emmarex/plantdisease. Accessed: 2025-09-17. 2025.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012)*. 2012, pp. 1097–1105. URL: https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[25] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

Appendix A

State of the art

A.1 Traditional Image Processing

Biswas et al. (2016) [10] used Fuzzy C-Means clustering in combination with a multilayer perceptron to assess blight severity in potatoes. The data consisted of 27 images captured by untrained farmers in India under varying lighting conditions, distances, and orientations. Their pipeline applied de-correlation stretching to enhance color differences before using Fuzzy C-Means to segment each image into healthy leaves, diseased leaves, and background. A multilayer perceptron was then used to classify the blight severity. The method achieved an accuracy of 93%.

A key strength of this approach is its explainability: the segmentation makes it easy to see which parts of the plant are considered healthy or diseased. However, the dataset is very small, which limits generalization, and the method is sensitive to background elements such as soil or shadows that share similar colors with blight symptoms. More broadly, traditional image-processing methods typically require less data than deep-learning models, but they are also more sensitive to environmental conditions.

Franceschini et al. (2019) [7] investigated how potato plants change spectrally when infected with late blight, using hyperspectral images collected from a UAV. Their method analyzed how the light reflected from plants varied over several flight days and linked these changes to disease severity. This allowed them to detect late blight at a very early stage, even when only 2.5–5% of a leaf was affected.

The authors also compared their results to patterns observed in ground-level images and found comparable performance between ground-based and UAV-based measurements. This demonstrates the potential of UAV-mounted hyperspectral sensors for early, field-scale blight detection. However, hyperspectral cameras are expensive and have limited spatial coverage compared to standard RGB systems, which makes large-scale deployment challenging.

A.1.1 Transfer Learning

Pardede et al. (2018)[22] attempted to classify potato diseases using RGB images. Their approach combined a convolutional autoencoder for unsupervised feature extraction with an SVM classifier trained on the learned latent representation. This reduces the reliance on large labeled datasets, since the encoder can be trained on unlabeled images, and only a smaller labeled subset is needed to train the final classifier. On potatoes, they achieved an accuracy of around 87% using a linear SVM kernel and 84% with an RBF kernel. The model was trained on the PlantVillage dataset, a large open-source collection of over 54,000 plant leaf images covering 14 crop species and 38 disease classes. While PlantVillage [23] has become a standard benchmark for plant disease classification, it has some important limitations. In particular, the dataset only contains images of single leaves photographed under controlled laboratory conditions with uniform backgrounds and lighting. This makes the classification task significantly easier than working with more complex data such as UAV imagery or whole-plant images in the field, where leaves overlap, backgrounds are heterogeneous, and lighting conditions vary.

As the transformer revolutionized natural language processing, Vision transformers where proposed for computer vision. Taking advantage of the attention mechanism ViT are able to capture global context and long range dependencies better due to the global attention mechanism. Espitalier et al. [11] employ a transfer-learning strategy built around Vision Transformers for high-resolution species detection. Their objective is to identify invasive and other plant species in roadside imagery, a task that requires the model to recognise small, fine-grained plant features across large high-resolution scenes. To achieve this, they start with a BEiT-based ViT model that undergoes several stages of extensive pretraining.

First, the model is pretrained in a self-supervised manner on ImageNet-22k, learning general visual representations without labels. It is then supervised-trained on ImageNet-22k, this time learning to classify 22,000 object categories. After this, the network is fine-tuned on ImageNet-1k, completing a multi-stage pretraining pipeline that provides very strong generic visual features.

Using this BEiT model as a starting point, the authors then further fine-tune it on the Pl@ntNet dataset, which contains 6,585,369 images across 43,683 plant species. This additional training stage adapts the generic BEiT model into a domain-specialized plant-recognition model capable of detecting subtle morphological cues. The result is a Vision Transformer already highly proficient in recognizing plant species before it is applied to the high-resolution roadside detection task.

As they are also attempting to predict rare species under severe class imbalance, they address this issue by selecting a prediction threshold using the validation set that maximizes the *balanced accuracy*, defined as

$$\text{Balanced accuracy} = \frac{1}{2}\left(\frac{TP}{P} + \frac{TN}{N}\right),$$

where $\frac{TP}{P}$ is the recall of the presence class and $\frac{TN}{N}$ is the recall of the absence class. This metric gives equal importance to both classes and prevents the model from achieving deceptively high accuracy by always predicting absence.

Espitalier et al. demonstrate that extensive pretraining substantially improves minority-class performance. The rare invasive species class of interest appears only 66 times out of 2,074 samples, yet the pretrained BEiT model achieves a balanced accuracy of 90.11%, compared to 82.73% without Pl@ntNet pretraining. This highlights a major strength of their approach: large-scale domain-specific pretraining enables the model to reliably detect extremely underrepresented classes that would otherwise be overlooked.

The main weakness is the cost of achieving this performance. Their pipeline relies on several stages of massive pretraining which requires computational resources far beyond typical research or applied settings. Thus, while the method excels at handling severe class imbalance, its dependency on extremely large datasets and heavy pretraining limits its accessibility and reproducibility.

Parez et al. (2023) introduce GreenViT, a lightweight Vision Transformer specifically designed for plant-disease detection. The model substantially reduces the size of a standard ViT-Base architecture—from 86M to only 21.65M parameters—while maintaining or even improving accuracy. The authors also investigate different transfer-learning strategies to assess how pretraining and fine-tuning affect disease-classification performance across three benchmark datasets: PlantVillage (PV), the Data Repository of Leaf Images (DRLI) and the Plant Composite (PC) dataset. On all three test sets, GreenViT achieves state-of-the-art performance, with average test accuracies of 0.9740% (PV), 0.9445 (DRLI)% and 0.9579% (PC). [12] Its strong performance on diverse datasets demonstrates that compact transformer architectures can generalize well to subtle visual symptoms—highlighting the potential of lightweight ViT variants for real-world agricultural imaging tasks, including UAV-based detection of early potato disease risk indicators.

Appendix B

Data and Pre-processing

B.1 Ortho-mosaic pre-processing

B.1.1 Rotation

To prepare the ortho-mosaics for annotation, each image was rotated to achieve a consistent horizontal alignment. This alignment simplifies the annotation process, as the imagery then follows the sprayer tracks, providing a point of view similar to how farmers visually inspect the field.

Using the Python library rasterio [1], a down-sampled thumbnail of each mosaic was first generated. A minimum-area (smallest enclosing) rectangle was then fitted to the valid image footprint, and the orientation of its long edge relative to the x-axis was used to estimate the required rotation angle. Finally, the full-resolution raster was rotated around its center in pixel space to produce a horizontally aligned ortho-mosaic.

B.1.2 Cropping

To crop the ortho-mosaic, we applied a similar procedure. First, the image was downsampled to reduce computational cost. A binary mask was then created by marking pixels with values greater than zero as valid. Based on this mask, the smallest enclosing rectangle containing all valid pixels was identified and its corner coordinates were extracted. These coordinates were subsequently rescaled to the full-resolution grid, and the original image was cropped accordingly, ensuring alignment with the underlying data.

B.1.3 Extracting tiles

Edge detector

During the creation of the orthomosaic, minor border distortions were introduced, causing the edges of the image to deviate slightly from a perfect rectangle. To correct for this, I

[1] `https://rasterio.readthedocs.io/en/stable/`

applied an edge-detection that identifies the valid inner region of the image. Starting from each side, the algorithm scans inward to find the first row or column in which at least 95% of the pixels are non-black. The intersection of these boundaries defines the largest inner rectangle that contains no black borders or distorted edges.

Saveing tiles

The extracted tiles were saved in NumPy's native .npy format. This format preserves both the array shape and data type, ensuring that the data can be reloaded without any loss of information or precision. Moreover, it allows for fast binary read/write operations, making it well suited for large-scale training pipelines in PyTorch.

Appendix C

Machine Learning background

C.1 Convolutional Neural Networks

In 1989, LeCun et al. proposed using CNNs for handwritten digit recognition. This demonstrated how CNNs could overcome a key limitation of traditional multilayer perceptrons (MLPs), namely that images contain thousands of pixels, which would require an MLP to learn a very large number of weights and biases in the first layer.

A core property of CNNs is shift invariance, meaning that the model responds similarly to a pattern regardless of where it appears in the input. In LeCun's case, this was crucial because a digit could appear anywhere in the image. In our case of risk-area detection, shift invariance allows the model to detect risk regions regardless of their spatial location in the image, assuming, however, that the relevant features are locally present rather than globally distributed.

Another important property of CNNs is that they exploit the high correlation between neighboring pixels. Pixels close to each other contain related spatial information, and convolutional layers can extract these local features by restricting the receptive field. This makes CNNs effective for detecting local patterns but also means they may struggle to capture global, long-range relationships without additional architectural components. [13]

Modern CNNs CNNs have evolved significantly since LeCun et al.'s pioneering work on handwritten digit recognition in 1998. A major breakthrough came with AlexNet, introduced by Krizhevsky et al. (2012) [24], which demonstrated the potential of deep CNNs on large-scale image classification tasks. The model contained approximately 60 million parameters and 650,000 neurons, and was trained on the ImageNet dataset, comprising 1.2 million high-resolution images across 1,000 categories.

AlexNet's success was largely attributed to several architectural and computational innovations. It employed the Rectified Linear Unit (ReLU) activation function to mitigate vanishing gradients, dropout regularization to reduce overfitting, and overlapping max-pooling to improve generalization.

Mathematical foundation of cnn Convolutional Neural Networks (CNNs) are designed to enforce shift, scale, and distortion invariance through three key architectural principles: local receptive fields,weight sharing, and spatial subsampling [13]. Each neuron in a convolutional layer receives input from a restricted spatial region of the previous layer, known as its receptive field. This locality allows neurons to capture spatially correlated features such as edges, corners, or textures.

To enforce translational invariance, CNNs apply learnable filters (kernels) that are convolved across the input image, the same set of weights and biases is reused at multiple spatial positions, greatly reducing the number of parameters compared to fully connected layers and enabling pattern recognition independent of absolute position. Mathematically, the convolutional operation (sometimes also know as the cross-corelation) for an input feature map x and kernel w is

$$(y * w)(i, j) = \sum_m \sum_n x(i + m, j + n)\, w(m, n) \tag{C.1}$$

followed by the addition of a bias term and a non-linear activation function

$$h(i, j) = \sigma\big((x * w)(i, j) + b\big) \tag{C.2}$$

The resulting feature maps are progressively abstracted through successive convolutional layers, each learning increasingly complex representations of the input data. from simple edges in the first layer to object parts and semantic structures in deeper layers. Pooling or subsampling layers further reduce spatial resolution, providing robustness to small shifts and deformations while limiting computational complexity. For a even deeper dive into CNNs we refer to Goodfellow et al. (2016) [25]

C.2 Background ResNet

Motivation and development

As convolutional neural networks are made deeper, they can encounter optimization difficulties. He et al. [14] demonstrated that simply increasing network depth does not necessarily improve performance as intuition might expect. In particular, they identified the degradation problem, where deeper networks exhibit higher training error than their shallower counterparts, even when overfitting is not the cause. This observation motivated the introduction of residual connections, which enable more effective optimization of deep architectures.

He et al. argue that if a shallow network can achieve a certain accuracy, a deeper network that contains it as a subset should be capable of at least the same performance, since the added layers could simply implement identity mappings. Yet, in practice, optimization algorithms fail to find such solutions efficiently.

To address this, they propose the residual learning framework, where the stacked layers learn a residual function:

$$F(x) = H(x) - x \tag{C.3}$$

and the final output is expressed as:

$$y = F(x) + x \tag{C.4}$$

In this formulation, x is the input to a residual block, $H(x)$ denotes the desired underlying mapping, and $F(x)$ is the residual function learned by the stacked layers. The output y is obtained by adding the input x to the learned residual via a skip connection.

This identity shortcut connection allows gradients to propagate more effectively and enables successful training of very deep networks. Thus reaping the rewards of very deep networks in terms of increase in accuracy.

Strengths and weaknesses

There are few disadvantages to introducing residual connections into CNNs. The addition of identity shortcuts primarily facilitates the training of deeper and more expressive architectures by improving gradient flow and optimization stability. However, one potential drawback is that larger and deeper models generally require more data to achieve optimal generalization. In our case, the available dataset is limited compared to large-scale datasets such as ImageNet, which may constrain the benefits that deeper residual architectures can offer. Another drawback is that it that the model down-samples quickly which can distort our ability to detect fine grained details like our features.

C.3 Mathematical Foundation of Vision Transformers

The Vision Transformer (ViT) operates on an image input $\mathbf{x} \in \mathbb{R}^{H \times W \times C}$, where H and W denote the height and width of the image, respectively, and C denotes the number of channels. The image is partitioned into a sequence of non-overlapping two-dimensional patches of size $P \times P$.

Each patch $\mathbf{x}_p^i$ is flattened into a one-dimensional vector and linearly projected (using a fully connected layer) into a latent embedding space of dimension D, resulting in a sequence of patch embeddings:

$$\mathbf{x}_{pE}^i = \text{Linear}\big(\text{Flatten}(\mathbf{x}_p^i)\big), \quad i = 1, \ldots, N, \tag{C.5}$$

where

$$N = \frac{HW}{P^2} \tag{C.6}$$

denotes the total number of patches.

A learnable `[class]` token $\mathbf{x}_{\text{class}}$ is prepended to the sequence of patch embeddings, and learnable positional embeddings $\mathbf{E}_{\text{pos}}$ are added to retain spatial information:

$$\mathbf{z}_0 = \left[\, \mathbf{x}_{\text{class}};\ \mathbf{x}_{pE}^1;\ \mathbf{x}_{pE}^2;\ \ldots;\ \mathbf{x}_{pE}^N \,\right] + \mathbf{E}_{\text{pos}}, \quad \mathbf{E}_{\text{pos}} \in \mathbb{R}^{(N+1)\times D}. \tag{C.7}$$

Each Transformer encoder block consists of multi-head self-attention (MSA) followed by a feed-forward multilayer perceptron (MLP), both surrounded by residual connections and layer normalization (LN):

$$\mathbf{z}'_\ell = \text{MSA}(\text{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \quad \ell = 1, \ldots, L, \tag{C.8}$$

$$\mathbf{z}_\ell = \text{MLP}(\text{LN}(\mathbf{z}'_\ell)) + \mathbf{z}'_\ell, \quad \ell = 1, \ldots, L. \tag{C.9}$$

Here, $\mathbf{z}_\ell \in \mathbb{R}^{(N+1)\times D}$ denotes the sequence of token embeddings after the ℓ-th Transformer encoder block, where the first token ($\mathbf{z}_\ell^0$) corresponds to the class token and the remaining tokens correspond to image patches.

The final representation used for classification is obtained from the class token after the last encoder block:

$$\mathbf{y} = \text{LN}(\mathbf{z}_L^0), \tag{C.10}$$

where $\mathbf{z}_L^0$ denotes the class token after the final layer.

Within each MSA module, the self-attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V, \tag{C.11}$$

where

$$Q = \mathbf{z}_{\ell-1} W_Q, \quad K = \mathbf{z}_{\ell-1} W_K, \quad V = \mathbf{z}_{\ell-1} W_V. \tag{C.12}$$

Here, Q (*queries*) represent what each token attends to, K (*keys*) describe what each token provides, and V (*values*) contain what information is used. The dot product $QK^\top$ measures pairwise similarity between tokens, which is scaled by $\sqrt{d_k}$ for numerical stability and normalized using a softmax function to obtain attention weights.

In multi-head self-attention, h independent attention heads are computed in parallel, allowing the model to capture different types of relationships between tokens. The outputs of all heads are concatenated and linearly projected back to dimension D.

For deeper insight into the background of Vision transformers we refer to Dosovitskiy(2021) et al. [16]

Appendix D

Receptive Field Analysis

This appendix provides a detailed derivation of the receptive-field optimized for detecting annotated risk regions in 224×224 tiles. A tile is labeled *High Risk* only if at least 5% of its area overlaps with a high-risk polygon. The goal of this analysis is to estimate the minimum receptive field needed for a model to reliably capture such regions.

For a 224×224 tile:

- **Total pixels:** $224 \times 224 = 50{,}176$
- **5% area:** $0.05 \times 50{,}176 = 2{,}508.8$ pixels

A region covering 5% of the tile can be interpreted in two equivalent ways:

1. **Square-equivalent side length:**

$$s = \sqrt{0.05} \cdot 224 = \sqrt{2{,}508.8} \approx \mathbf{50.1} \text{ px}$$

   Corresponding to a roughly 50×50 pixel patch.

2. **Disk-equivalent diameter:**

$$d = 2\sqrt{\frac{0.05 \cdot 224^2}{\pi}} \approx \mathbf{56.6} \text{ px}$$

   Corresponding to a roughly 57 pixel circular region.

A receptive field that only matches the minimal region would therefore lie in the range of 50–57 pixels. In practice, additional margin is typically required to capture contextual shape, texture, and local neighborhood information surrounding the annotated region. A practical receptive-field target for this task is therefore in the range of approximately 60–120 pixels.

Appendix E

Classification results

(a) False Positive tiles. These are tiles annotated as *No Risk* in the ground truth but predicted as *Risk* by the model. Many of these tiles contain structural irregularities such as shadows, uneven canopy, or exposed soil, which may visually resemble symptoms of plant collapse.

(b) True Negative tiles. These examples show healthy potato plants correctly classified as *No Risk*. The canopy structure is generally uniform, with no visual signs of collapse, discoloration, or missing plants.

Figure E.1: Examples of model classifications from the test set. False Positives (left) illustrate visual patterns that may confuse the model, while True Negatives (right) show healthy plants exhibiting no detection-relevant symptoms.